1-1-2003

Development, implementation and evaluation of an interactive multimedia instructional model : A teaching and learning programming approach

Kacha Chansilp
Edith Cowan University

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

Development, Implementation and Evaluation of an Interactive Multimedia Instructional Model:
A Teaching and Learning Programming Approach

By

Kacha Chansilp

B.A.(Computer Science: Queens College, NY)

M.A.(Graphics Design (Hons.): NYIT, NY)

A Thesis Submitted in Fulfilment of the Requirements for the Degree of
Doctor of Philosophy (Interactive Multimedia Technologies)

School of Communications and Multimedia
Faculty of Communications and Creative Industries
Edith Cowan University, Perth, Western Australia

May 30, 2003

ABSTRACT

This study sought to explore the outcomes from the use of a dynamic interactive visualisation tool among novice programmers in an introductory computer programming course. The proposed model, Dynamic Interactive Visualisation Tool in Teaching C (DIVTIC), was designed to use multimedia and visual imagery to provide learners with a step-by-step representation of program execution in the C language as a means of enhancing their understanding of programming structures and concepts.

DIVTIC was designed to support constructivist learning principles and combined collaborative and visualisation learning strategies with use of the Internet and the World Wide Web to support the learning of programming. The feasibility and effectiveness of DIVTIC was explored among a cohort of 100 undergraduate engineering students, 50 in a control group and another 50 in an experimental group, studying an introductory programming course at Suranaree University of Technology (SUT) in Thailand.

The study found that the use of DIVTIC was a successful complement to conventional teaching. The results clearly demonstrated the advantage of using DIVTIC among low achieving students. The students from this level in the experimental group significantly outscored their counterparts in the control group in the final test suggesting that DIVTIC was an important element in their learning process. Interestingly, these low achieving students used DIVTIC most and achieved highest grades. However, lower achieving students appeared to learn from simply viewing the animations rather than being highly interactive and stopping and starting them consistently. The study found that the visualisation process implemented in DIVTIC could be of considerable assistance to a particular group of students, those with a low GPA, in developing their understanding of difficult programming concepts.

DECLARATION

I certify that this thesis does not, to the best of my knowledge and belief:

(i) incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education;

(ii) contain any material previously published or written by another person except where due reference is made in the text; or

(iii) contain any defamatory material.

Signature: [redacted]

Date: 30 - 5 - 03

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to many people who made this thesis possible. Without these people, I would not have had the opportunity to write this statement and be able to submit this thesis.

Firstly, my special thanks will go to Prof. Wichit Srisa-an, Founding Rector of Suranaree University of Technology (SUT), Thailand, who helped to support me from different sources including Suranaree University of Technology and the Faculty of Communication, Health, and Science at Edith Cowan University (ECU), Western Australia, with a full time scholarship that made it possible for me to undertake this study.

Secondly, I am deeply indebted to my supervisor, Professor Ron Oliver, and associate supervisor, Dr. Arshad Omari, for their invaluable and impressive guidance, nurturing, encouragement, and support in every stage of my study. Their profound knowledge, kindness, patience, vision, and valuable hints have been provided me with the ability to complete and enrich the thesis with lifetime benefit that will be unforgettable.

Thirdly, I am deeply grateful to:

- Assoc. Prof. Jim Millar, Director at Learning and Development Service Centre, ECU; and
- Prof. Craig Standing, Foundation Professor at Business and Public Management, ECU

for reviewing my initial proposal and for providing very useful comments and guidance.

Fourthly, I would like to thank the support teams from SUT including:

- Asst. Prof. Suyut Satayaprakorb, Head of School of Computer Engineering;
- Dr. Tanongsak Bisamsin, Director of Center for Computer Services;
- Mr. Sompan Chansilp, Computer Programming in C instructor; and
- All tutors and students who participated in the 408101 course, trimester 2/2001

and from ECU including:

- Dr. Danielle Brady and Dr. Susan Hill, Research Consultants;
- Dr. Judith Rochecouste, Academic Writing Consultant;
- Ms. Michelle de Souza and Ms. Loraine Dominic, Graduate Research Assistants;
- Ms. Sharron Snader, Executive Assistant;
- Ms. Lyn Leslie, Library Account Manager;
- Mr. Paul Halfpenny, Post Graduate Information Officer;
- Ms. Catherine Samananda, International Project Manager, Thailand;
- Mr. Russell Miller and Mr. Steven Doyle, Media Support Officers; and

my special thanks goes to Dr. Jan Herrington for her kindness in sparing her invaluable time to looked through the thesis and give useful suggestions.

Last, but not least, I would like to give very special thanks and all my love to my family. I could not have survived the long period of four years in Perth and been successful without the understanding and unrelenting support, patient love, and caring of my lovely wife—Yupasri Chansilp, my good son—Chanoot Chansilp, and my pleadingly talkative daughter—Natcha Chansilp. Without them, I doubt that this thesis would ever have been written. I also apologise to them for the anxiety which must have been caused by my absence. I now wish to pay them back by sharing the delight of this accomplishment together and forever.

I dedicate this work to my parents, Mr. Sompong and Mrs. Kimbouy Chansilp. Unforgettable thanks also go to my oldest brother, Mr. Panya Chansilp, who encouraged me to go to a university in New York as a first step towards this achievement. I am deeply grateful for this invaluable support and encouragement in those early stages and although I cannot repay this kindness, the memory of it will be in my mind and forever.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

As the impact of technology grows, interactive multimedia is being used more and more in the educational sector. This means that researchers need to optimise the use of new technologies in teaching and learning environments. Thus, instructional materials and tools are being produced by incorporating updated software with advanced delivery technologies such as the Internet (e.g., Oliver, Herrington, & Omari, 1996; Rowe & Thorburn, 1999; Warendorf, 1997; Yoo, 1998).

The teaching and learning of a programming languages is a difficult task (Daly, 1999; Gray, Boyle, & Smith, 1998; Hagan & Lowder, 1996; Jehng & Chan, 1998). Selby and Young (1998) state that computer programming courses play an important role as gatekeepers for further studies in Computer Science. So first year students need a skilled teacher who has experience in teaching computer programming at that level. Conventional instruction may not always be appropriate to help students in developing semantic knowledge (Oliver & Malone, 1993) even though there have been many efforts to create tools for the teaching and learning of programming languages (e.g., Daly, 1999; Jehng & Chan, 1998; Rowe & Thorburn, 1999; Smith & Webb, 1998). This still remains a problem in many institutions (Carter & Jenkins, 1999).

New programming languages tend to add complexity to the task because they have more advanced features. For example, for many years, Pascal was the major introductory computer programming language and was the most popular first language for teaching students in the nineties (Brilliant & Wiseman, 1996). However, Pascal is no longer used in industry (Hubbard, 1996) and many institutions have now switched from Pascal to C. The problems experienced by novice programmers learning a procedural language such as Pascal as their first programming language increase when learning C (Hubbard, 1996; Smith & Webb, 1998). The C language is difficult for many novices to learn on their own (Johnson, 1995).

This study sought to explore how contemporary multimedia technologies may be used to enhance the teaching and learning of these programming languages.

1.1 The Background to the Study

Computer programming courses are more difficult and time consuming than other courses for the majority of students (Hagan & Lowder, 1996). Usually, novices have different background knowledge before entering an introductory computer programming course, meaning that a class may contain students with a wide range of proficiency levels. Wilcocks and Sanders (1994) suggest the use of a computer-aided dynamic program animator to support the instruction of weaker students who may be too shy to ask questions or who have difficulty gaining attention when classes are large. Such programs can provide more opportunities for weaker students because they can be used over and over again.

Bishop-Clark (1995) states that computer programming is an intricate task, which can be divided into four different phases: (a) problem representation, (b) design, (c) coding, and (d) debugging. The requirements for each phase differ in terms of cognitive style and personality traits which affect the design stage significantly more than the coding stage (Bishop-Clark, 1995). This suggests the optimal role for a technology solution might be in the design stage.

Generally, students differ in their ability to understand material which is very abstract and difficult to visualise. Previous research has proposed ways to improve instructional materials and therefore student outcomes. For example, instruction can incorporate a dynamic explanation tool to help students visualise each step in program execution (Karsten & Kaparthi, 1998; Lischner, 2000; Rowe & Thorburn, 1999). New technologies now provide many more options for the designer and can be used to support the teaching and learning process.

Technology provides many ways of improving instructional materials to help instructors improve the teaching and learning environment. This research sought to develop an instructional model using the visual capabilities of the most recent developments in technology to explore ways of enhancing the teaching of introductory programming.

1.2 The Significance of the Study

The number of computers in use worldwide is increasing rapidly and by end of the year 2000 was estimated to be approximately 579 million (*Computers-in-use in year 2000*, 2000), the demand for good programmers is becoming important in our society. McKeown and Farrell (2000) state that there will be a shortage of qualified computer programmers in the near future. This will have major consequences for education and for the global economy.

Students in science majors, especially in the computing, need to learn valid programming concepts during their introductory computing courses because these form a strong background for more advanced programming courses in their university curriculum (Herrmann & Popyack, 1994). At the moment, much of the teaching is based on textbooks and this does not always work well. Many students who finish introductory classes, are still weak in their understanding of basic concepts. Kann, Lindeman, and Heller (1997) claim that the graphic representation of algorithms used in most textbooks are abstract visualisations and not sufficient for learners to develop logical thinking skills required in computer science courses. Students' problems are mostly based on the lack of conceptual understanding and mental models (Soloway, Ehrlich, Bonar, & Greenspan, 1981). This provides an opportunity to investigate ways to enhance learning through the informed use of contemporary graphics programs.

The research presented in this thesis sought to explore the opportunity for creating a technology based learning model that used visualisation to provide student activities for building their understanding of the programming process. The model has potential benefit instructors who want to move from traditional instruction methods to using technology as a teaching medium.

This research sought to explore not only how to help students to learn effectively, but also how to:

- prepare students with a strong background in computer programming;
- reduce learning time;
- save teacher consultation time, and
- engage students in student-centred learning.

This research should help to provide useful support and resources to ensure that there will be a supply of skilled programmers to serve our future economic needs.

1.3 The Purpose of the Study

The purpose of this study was to explore how students in introductory programming courses were aided by the use of interactive visual and Web-based instructional materials. This study used the opportunities of new technologies to create a learning environment that could help to increase the learners' understanding. A model was designed to help students construct valid mental models of computer programming. It sought to provide a basis for learning in any programming language. In this study, the C language was selected for the following reasons:

- C is a popular language that is taught in many institutions and is sufficiently widespread in industry (Newlands, 1992); and
- This study was conducted at Suranaree University of Technology (SUT) in Thailand, which uses C as its introductory programming language.

The novelty and originality of this study is in the innovative use of graphics and interactivity in computer-based learning and the exploration of how these materials influence students' learning outcomes. Another novel feature has been the use of the materials in a Thai university where students learn programming using English language statements and control structures, despite poor spoken and written English skills. The impact of the different language requirements was an important aspect of the inquiry and another distinctive element of the research.

1.4 The Organisation of the Thesis

The thesis is organised into eight chapters which are described as follows:

- Chapter 1 provides the background, significance, and purpose of the study;
- Chapter 2 presents a review of literature relevant to this research including the problems and difficulties in learning to program, strategies in teaching and learning programming, previous findings relating to professional development models and their evaluation, and technology support;

- Chapter 3 describes the conceptual framework for the study. The methods of visualisation, collaboration, and constructivism are used to explore how instructional design can be used to benefit teaching and learning in introductory computer programming. This chapter concludes with research questions and an overview of the interactive instructional model which was designed, developed, implemented, and evaluated in the teaching and learning setting;
- Chapter 4 contains a description of the development of DIVTIC (Dynamic Interactive Visualisation Tool in Teaching C) which was based on the contemporary learning theories applied using current communication technologies. Each component of DIVTIC is described. The pilot study which was conducted to determine the feasibility of some of the components of DIVTIC, and the subsequent modifications, is also described;
- Chapter 5 begins with a literature review of the research methodology used in the study followed by the particular methods used in the study. The reliability and validity of the data collection are also discussed;
- Chapters 6 and 7 present the data collection, explanation, analysis, and address each of the research questions. Chapter 6 explores how students used DIVTIC, while chapter 7 explores to what extent this experience influenced students' performance; and
- Chapter 8 presents a discussion of the problems encountered during the study, the conclusions of the findings, the limitations of the study, and the potential for further research.

The review of the related literature is divided into four general areas:

- The problems and difficulties that students face when learning to program;
- The strategies that teachers use to teach programming;
- How students learn to program; and
- How technology can support teaching and learning environments.

2.1 The Problems and Difficulties that Students Face When Learning to Program

Computer programming is an area that contains complex knowledge and abstract concepts that need individual mental effort to learn and understand (Jehng & Chan, 1998). Learning computer programming involves several cognitive abilities including syntactic knowledge, conceptual knowledge, and strategic knowledge (Bayman & Mayer, 1988). This study focuses on the problems and difficulties of learning to program, in particular, (a) difficulties in learning syntax; (b) difficulties in learning semantics; and (c) difficulties in acquiring strategic knowledge.

2.1.1 Difficulties in Learning Syntax

Syntactic knowledge refers to knowledge of lexical units, which consist of specific details and rules, such as knowing that the end of each statement in C must end with a semicolon (Bayman & Mayer, 1988; Fay & Mayer, 1994; Oliver & Malone, 1993). Lischner (2000) states that syntax in a programming language has complex rules and is difficult to learn and understand. Novices must learn commands and statements in relation to the syntax of the language.

Natural languages, like English, have rules which are much more flexible than those of a computer programming language. People are able to work out what you are saying although you may not be grammatically correct. In a programming language, if you give the incorrect syntax, the computer cannot understand what you are trying to do. Mayer and Fay (1987) note that students tend to use their intuition from their understanding of

natural language when programming commands. For example, with the compound *If* statement, students may write a code as:

If A > B and C Then ...

In this case, novice students may assume incorrectly that "the comparisons are being made between A and B and A and C" (McKeown & Farrell, 2000, Focus of programming research, para. 4), when in fact this is a chain of evaluation based on operator precedence.

du Boulay (1986) states that the keywords used in a programming language sometimes conflict in meaning. For example, the "repeat" statement can sometimes mislead novices who think that something needs to be repeated, but this is not the case. Another example by du Boulay (1986) is in the use of the boolean operator "and" which can mislead novices to think in the sense of "what is next" such as "wash your hand and eat your food." Novice programmers also are confused with the assignment of variables and arrays which they usually understand in terms of mathematics. For example, in C we can say A = 2 but not 2 = A, which is syntactically invalid.

Research in this area shows quite clearly that many novice programmers have trouble learning the syntax of programming languages. One of the principal reasons seems to be the conflict with natural language. The syntax or keywords used in a programming language sometimes conflict in meaning. For novice programmers with poor English, such as in Thailand where English is a foreign language, this problem can be heightened. This suggests a need for more research to explore ways to explain these problematic concepts and to help students to understand the syntax of programming languages. One possible solution that can overcome these problems is the use of a teaching and learning environment that provides immediate feedback (e.g., Alam & Renci, 1998), a syntax aware program editor.

2.1.2 Difficulties in Learning Semantics

Semantic knowledge refers to the action that occurs in the computer in response to a given instruction such as adding and deleting (Fay & Mayer, 1994). Novices often assume that the computer will understand and be able to execute their incomplete commands because they can understand them (Perkins, Schwartz, & Simmons, 1988).

Students therefore have difficulty adapting their existing skills when writing programming code (McKeown & Farrell, 2000). Their previous understanding will interfere with some commands in programming. For example, in mathematics variables or constants on both sides of an equal sign mean that they are equivalent. In programming code, however, students may have a problem when they see a code like

```
Total = Total + 1;
```

because there is no such number that equals itself when one is added to it. Similarly, when programming in BASIC, novices are often confused when they see an assignment statement such as LET A = A + 1 because they think that the "A"s on both side of the equal sign have been treated in the same way. They do not understand the sequential nature of program execution where "one stands for a location and the other for a value" (du Boulay, 1986, p. 64).

The most common example in C which supports du Boulay's comments is the swapping of a value between two variables (e.g., A and B) where we need to use a temporary third variable (e.g., TEMP). Novice programmers will usually swap the value between variable A and B without using the third temporary variable, TEMP. For example, suppose that the value of A is 5 and the value of B is 7, they will write this as follows:

```
A = B;
B = A;
```

which appears to be correct, but is actually incorrect programming. After the processing the first statement, the value of variable A is changed from 5 to 7 but the value of variable B still has the same value, which is 7. In the following statement, the value of variable B is not changed since the value of variable B is 7 and the value of A is now 7. Therefore, both variables A and B have the same value of 7. As du Boulay (1986) states, a variable can hold only one value, thus we need to use the third temporary variable, for example, TEMP, so given that A = 5 and B = 7:

```
TEMP = A;
A = B;
B = TEMP;
```

After processing three statements, the variable TEMP and B have the value of 5, since A has the initial value of 5; the variable A will have the value of 7, since B has an initial

value of 7. This example provides evidence that students have difficulty in understanding some programming concepts. They need to visualise what is actually happening inside of the computer memory when each statement of the program is executed.

Research by Soloway et al. (1981) shows that novice programmers misunderstand the use of appropriate control statements (i.e., *for, while, repeat... until* loop). Students in an introductory Pascal programming class were given a simple problem that involved reading data, looping, testing and operating on a variable, and then writing the output as follows:

> Write a program which repeatedly reads in integers until their sum is greater than 100. After reaching 100, the program should print out the average of the integers read in (Soloway et al., 1981, p. 28).

Only 44% of students were able to write the correct code. While many of students made no particular errors in their programming, those that did included two types of errors: (a) syntax and semantics, and (b) deciding which constructs to use and how to operate them.

From research by Soloway et al. (1981), both novices (enrolled in introductory Pascal programming class), and intermediates (enrolled in a second course in programming using Pascal, i.e., data structures course) showed the same common problems when using the loop construct. The subjects tended to use *while* loops in all situations rather than *for* loops to solve problems.

Novice programmers also had problems with the conflicts between the update of a counter variable, which counts the number of loops undergone, and the update of the running-total variable, which accumulates all the supplied variables. For example, they misunderstood a particular type of assignment statement by using the pattern for updating the counter variable (i.e., I := I + 1), for which a constant is required, when they wanted to update the running-total variable and a variable is required (Soloway et al., 1981).

The literature shows that the acquisition of such semantic knowledge is significantly difficult for novices. Commands in a programming language are not the same as in the everyday use of the English language and mathematics. Rather, they must be specific

and follow exact rules of the programming language so that they can be compiled and executed correctly. Novices need to be able to visualise what is actually happening inside the computer memory when each statement of a program is executed. This supports the need for more research into finding ways to provide students with a better understanding of a programming language, for example, by adding imagery to the teaching and learning of programming.

2.1.3 Difficulties in Acquiring Strategic Knowledge

Strategic knowledge, or transfer strategies, refers to techniques to plan and combine syntactic and semantic knowledge when constructing a program to solve a given problem (Bayman & Mayer, 1988; Fay & Mayer, 1994). Soloway (1986) states that the real problem for novices is how to compose a chunk of syntax which will solve the given problem.

Bayman and Mayer; Fay and Mayer; Goei and Pieters; Mayer; Perkins, Schwartz, and Simmons; Sleeman, Putnam, Baxter, and Kuspa; and Sloane and Linn (cited in Shih & Alessi, 1993-1994, p. 157) assert that "research has found that misconceptions appeared to be one of the biggest obstacles students face in learning to program." Mulholland and Eisenstadt (1998) also state that novices have more difficulty understanding what an execution is doing rather than the algorithm design and planning. Cardinal (1991) attributes difficulty in interacting with a computer to inadequate understanding of it:

> Conceptual models bridge the gap between a computing system and the user's mental representation of that system. Too often, users of interactive devices, such as microcomputers, acquire mental representations derived from poorly organized and misunderstood interaction with the computer. The learner's reactions to a computer system may even appear to border on the superstitious. These computer users are unable to cope with minor system crises. This results from the formation of inadequate mental models. (p. 163)

The majority of novices create a faulty mental model (Shih & Alessi, 1993-1994). Johnson-Laird (1983) explains these mental models as follows:

> Understanding certainly depends on knowledge and belief. If you know what causes a phenomenon, what results from it, how to influence, control, initiate, or prevent it, how it relates to other states of affairs or how it resembles them, how to predict its onset and course, what its internal or underlying 'structure' is, then to some extent you understand it. The psychological core of understanding, I shall assume, consists in your having a 'working model' of the phenomenon in your mind. If you understand inflation, a mathematical proof, the way a computer works, DNA or a divorce, then you have a mental

> representation that serves as a model of an entity in much the same way as, say, a clock functions as a model of the earth's rotation. (p. 2)

The problem is also caused by the change processes inside the computer. Novice programmers do not fully understand these because they cannot see what is going on inside the computer (du Boulay, 1986; Mulholland & Eisenstadt, 1998). For example, when giving students a problem in BASIC that printed a square of stars where the user inputed the number of stars on each side, it was evident that some students had not previously encountered a repeated loop that crossed a single line of output (Perkins, Hancock, Hobbs, Martin, & Simmons, 1986). This is because students did not visualise what was happening when the program is executing.

The literature shows that novices have difficulty in constructing syntax and understanding the semantics required to solve a given problem. They need better conceptual knowledge to help them solve programming problems. This literature appears to support the need for further research on strategies to develop strong conceptual knowledge when learning how to program. To overcome this problem, a teaching and learning technique that enables students to construct strong strategic knowledge is needed. The use of a visualisation tool which enables students to see the internal workings of program execution would address this conceptual problem (e.g., Smith & Webb, 1998). Such a teaching tool could access knowledge at multiple levels of abstraction (Bergin et al., 1996).

2.2 The Strategies that Teachers Use to Teach Programming

In the past, most teaching styles have been teacher-centred, where teachers divide the content into small manageable modules and teach according to a prescribed lesson plan (Norman & Sphorer, 1996). These traditional teaching methods are lecture-based using static media which is often not well suited to conveying dynamic concepts (Jenkins & Towle, 1997; Wilcocks & Sanders, 1994). They can only convey the basic idea of syntax and semantics which is often not enough for novices to use a language or apply it to solving problems (Jenkins, 1998).

However, there are a number of computerised aids that are sometimes used in laboratories, but these are usually commercial debuggers designed for expert programmers and not suitable for novices (Smith & Webb, 1998). Most educational

programming packages have been designed to help students directly debug their programs rather than understand how they work (Itoh, Konishi, & Suzuki, 1998). A necessary requirement for novices is the development of appropriate schemata and mental models of the programming process (Jehng & Chan, 1998; Smith & Webb, 1998). This need is addressed by this study and investigates more dynamic ways to teach the syntactic, semantic, and conceptual requirements of programming.

2.2.1 Teaching Syntactic Method

Some of the dynamic teaching tools that have been developed include that of Daly (1999) who designed an online program submission and correction system called "RoboProf" to teach a subset of the syntax and semantics of C++. The package comprises of an outline course in HTML format with small programming problems on each topic which include if statements, loops, arrays, and strings. When a student submits an exercise, the system automatically marks it and provides immediate feedback which encourages the learning process (Alam & Renci, 1998). The system also shows the correct answer if the student makes mistakes. The student then has to return to the instructional material and submit a similar completed problem. Results have shown that all students pass the online course (with a 40 percent pass rate) and three-quarters achieved 90 percent or more. Daly's online course provided opportunities for students to increase their motivation and "improve their programming skills by completing gradually more difficult tasks" (Daly, 1999, p. 157).

2.2.2 Teaching Semantic Method

In the area of teaching semantic method, du Boulay (1986) proposed that program templates, with selected parts for novice programmers to fill in, help with learning to write programs. This has been successful because of the large cognitive load novices experience when learning a programming language. With only parts of a program to fill in there is not too much information for the novice.

Dyck and Mayer (1989) have suggested teaching semantics via native language (e.g., English) before teaching syntax will improve students' learning outcomes. Their research shows that students taught with corresponding English statements learn faster and more correctly.

Smith, Cypher, and Tesler (2000) have developed a Java-based authoring tool called "Creator" that allows users to create a personalised world and then animate it as desired without any knowledge of programming and without seeing a single line of program code. With this program, parents, students, and teachers can create interactive simulations, models, games, and demonstrate algorithms on the computer. Their research show that students who had used Creator shown more development in their learning outcomes than others without this experience.

2.2.3 Teaching Conceptual Method

Often, textbooks for computer programming languages are simply manuals for experienced programmers and inappropriate learning resources for novices (Segal & Ahmad, 1993). They generally only focus on syntax and semantics (Soloway, 1986). Research by Segal and Ahmad (1993) has shown that combining text and working examples in the instructional material can improve students' outcomes. They also state that the working examples must cover all aspects of instruction.

Teaching conceptual models which represent the states and relationships in programming can improve problem-solving performance and the development of better mental models (Bayman & Mayer, 1988; Shih & Alessi, 1993-1994). Bazik, Tamassia, Reiss, and Dam (1998) argue that algorithm animation and program visualisation tools can improve students' understanding of concepts. An example of this is VINCE, a C visualisation tool designed to assist novice programmers in understanding how the individual steps of a program are executed and used in memory. VINCE is written in Java and can be executed via a web page. It allows only syntactically correct C code to be traced, one operation at a time. The effect of VINCE has been studied by Rowe and Thorburn (1999) using 16 first-year students completing an introductory programming course in C. The students were divided into two groups of 8 with equal ability (i.e., a test group using VINCE and a control group not using VINCE). Evaluations showed no significant difference in students' perception of their programming ability. However, students in the test group showed improved comprehension in C. This improvement was attributed to the visualisation of each step of program execution.

Another such tool, TurtleGraph (Jehng & Chan, 1998), is a visual learning software product based on LISP-LOGO, which can be used to enhance student's understanding

the concept of recursion to solve geometric pattern drawing problems. This system features three instructional principles: (1) a reflective learning principle, (2) a reactive learning principle, and (3) a structured learning principle. The system provides an opportunity for learners to collaborate with each other in solving or discussing difficult tasks and comparing each other's code. The learners can observe the execution of their programs as they run. It also provides crucial examples to assist learners in solving problems.

The effect of TurtleGraph has been studied by Jehng and Chan (1998) using ninety-four social science students with no previous experience in computer programming. The students were divided into three learning environments: (a) distributed learning, (b) face-to-face learning, and (c) individual learning. Student performance in program generation was significantly different between the two collaborative learning conditions, (i.e., distributed learning and face-to-face learning) and the individual learning condition, $p < .05$. This study shows that collaborative learning conditions can improve student outcomes.

The teaching of computer locations (e.g., memory location) to novices can also be improved with online teaching tools. Smith and Webb (1998) have developed a low-level program visualisation tool, called Bradman, to assist novice programmers in seeing the internal workings of C program execution. This enables learners to develop their mental models and structural knowledge of the programming process in a progressive manner. Smith and Webb (1998) evaluated Bradman by using it with a group of twenty-four volunteer students who were taking an introductory programming unit. Half of the students, the test group, had access to Bradman, and the other half, the control group, did not. Results showed that the test group performed significantly better outcomes than the control group, which suggests that a visualisation tool can enable novice programmers to enhance their mental models of programming by providing different views of program execution.

Karsten and Kaparthi (1998) propose that using dynamic, visual teaching tools and resources via the World Wide Web (WWW) can help students develop appropriate mental models. They have developed a dynamic tool called "Web-based dynamic explanation" by using inexpensive software such as Microsoft PowerPoint to incorporate colour, graphics, animation, and narrative. This tool enables students to

visualise programming construction in C. Karsten and Kaparthi's research into the impact of this teaching tool suggests that, although development is a time-consuming process, the benefits are evident in the time saved in teaching and re-teaching concepts to novice programmers. Their research also indicates that visual explanations on the WWW are the most significantly helpful teaching aid ($M = 4.77$, $SD = 0.43$, $p < 0.000$) compared with examples on the board ($M = 4.30$, $SD = 0.79$), overheads and handouts ($M = 3.93$, $SD = 1.05$), IDE program debuggers ($M = 3.20$, $SD = 0.71$), and course textbook exercises ($M = 3.01$, $SD = 0.82$). This study, therefore had the same positive outcome as VINCE (Rowe & Thorburn, 1999).

In summary, the literature provides a number of reasons why students have difficulties in learning to program, for example:

1. Programming languages have strong conceptual bases that require considerable engagement and thinking, e.g., Jehng & Chan (1998);
2. Students do not understand the learning process. In fact, their learning processes are usually shallow, e.g., Smith and Webb (1998);
3. Students frequently learn aspects of programming separately so they do not understand the connection between various elements, e.g., du Boulay (1986); and
4. Students concentrate on syntax more than semantics, e.g., Dyck and Mayer (1989).

Learning a programming language in Thailand is even more difficult because English is not the main spoken language. Also more and more students are coming to university with low formal reasoning skills, meaning these students need more help in learning to program. A possible solution to overcome these problems is to provide students with a tool that will help them to understand the step-by-step programming process and to visualise program execution. This could help them to understand the sequential nature of program execution.

2.3 How Students Learn to Program

Race (1996) divides the learning process into four primary processes:

1. wanting to learn (motivation, thirst for knowledge);

2. learning by doing (practice, trial and error, getting one's hands dirty);
3. learning from feedback (other people's comments, seeing the results); and
4. digesting (making sense of what has been learned; getting a grip on it).

Race (1996) also proposes optimum conditions for learning:

1. at one's own pace;
2. at the time and place of one's own choosing;
3. with other people around, especially fellow-learners; and
4. when one feels in control of their learning.

This suggests that learning is a complex process and needs further exploration, especially with regard to learning to program. Learning to program is challenging in a number of ways and these challenges need to be recognised for novices (Perkins et al., 1986). Perkins et al. state that "learning by discovery is entirely appropriate considering the open-ended character of programming problems" (p. 40).

However, in most educational environments, students have become accustomed to conventional forms of teaching and learning where teachers are experts delivering knowledge to learners, who are passive receivers (Baldwin & Macredie, 1999; Oliver, 1999a). However, research on the philosophy of learning and teaching environments has discovered strategies which have more potential than traditional teaching styles (Gray, 1997; Norman & Sphorer, 1996).

There are many theories looking at how learning occurs and many examples to guide and inform teachers and learning designers. In developing materials and activities for learning to program, we need to explore those learning theories that promote higher-order learning and to explore learning strategies that are relevant and appropriate. To this end, an exploration of the theory of constructivism, and learning strategies such as collaboration and the use of visualisation follows.

2.3.1 Knowledge Construction

Previously, behaviorism was used as a theory to describe how learning occurs (Shim, 1998). Behaviorism is a learning theory which is based on the change in behavior of an organism. However, it is only concerned with what learners do and excludes the role of mental operations and activities in the learning process (Fardouly, 1998a; Jonassen,

1991). It is a theory in which teachers provide students with new knowledge to be memorised and repeated without providing an opportunity for them to make sense of the information they have been given. Zakari (1998) points out that behaviorists made a vital mistake in their theory by excluding the role of mental operations. Nonetheless, behaviorist theory dominated concepts of learning for most of the first half of the 20th century (Jonassen, 1991).

Nowadays, an alternative learning theory called "Constructivism" is widely followed in many educational institutions (Holzer, 1994). It is a philosophical theory of knowledge which argues that learners become active participants in constructing meaning and knowledge through experience, rather than through the passive reception of information (Ertl & Kraan, 1997; Lorsbach & Tobin, 1992; Zakari, 1998). This theory is concerned with internal mental states. Stephens (n.d., para. 2) states that "aspects of constructivist theory can be found among the works of Socrates, Plato, and Aristotle (ranging from 470-320 B.C.), all of which speak of the information of knowledge. However, the main philosophy of constructivism is generally credited to Jean Piaget."

Constructivist learning is active learning which McKinney (1996) describes as techniques where a learner does more than merely listen to a lecture, but also discovers, processes, and applies information. It occurs when the learner connects their past experiences with new information to form new knowledge (Butcher, n.d.). It is a situation where the learner is given tasks, information resources and opportunities to construct his/her own meaning and knowledge from the learning process (Fardouly, 1998b; Gagnon & Collay, n.d.). It involves the active creation and modification of thought, ideas, and understanding as a result of the experiences that occur within a socio-cultural context (Doolittle, 1998). Students themselves may have difficulty recognising their own existing knowledge since they believe that the teacher is an expert and it is his/her role to transfer knowledge to them. Thus, to successfully apply constructivist learning both teachers and students have to adapt to the model (Lorsbach & Tobin, 1992).

When the theory of constructivism is applied to science teaching, teachers change themselves from transmitters, who transfer knowledge, to facilitators who assist and encourage learning (Stephens, n.d.). Thus, the notion is that knowledge cannot be transferred from one person to another, but resides within individuals (Enerson,

Johnson, Milner, & Plank, 1997; Lorsbach & Tobin, 1992). Knowledge acquisition is the process of building accurate internal models by reconstructing and reorganising old knowledge in the light of new experiences (Doolittle, 1998).

An example of how constructivism can be applied in computing comes from Hagan and Sheard (1998). They claim that the addition of a one-hour class discussion between the traditional computing course sessions of a two-hour lecture and a two-hour laboratory session gives students the opportunity to discuss programming concepts and work with each other. It also bridges the gap between the lectures and the laboratory sessions. Results show that the discussion classes have greater potential and value for learning how to programme, because learners actively discuss and construct their own knowledge. Similarly, Jenkins (1998) proposes a change in students' participation from being "passive recipients of the teaching into active participants in a learning process" (p. 125). Jenkins also supports a participative approach to learning a programming language by offering additional sessions for students who have difficulty. His results show an improvement in students' learning outcomes and show that when students have the opportunity to interact with their peers, they can construct a better understanding of a subject.

The literature on constructivist learning shows that it has greater potential for improving learners' outcomes than traditional learning theory or behaviorism (Cobb, 1999; Hagan & Sheard, 1998; Jenkins, 1998). The constructivist model leads to an emphasis on the learning process rather than the teaching process. Therefore, it is student-centred, not teacher-centred. It provides opportunities for learners to be active in the learning process. The contrast between behaviorism and constructivisim parallels the African proverb: "If a man is hungry you can give him a fish, but it is better to give him a line and teach him to catch fish himself" (Papert, 1993, p. 139).

This literature suggests some significant benefits of knowledge construction such as:

1. engaging the learners and making the learning meaningful e.g., Ertl & Kraan (1997) and Zakari (1998);
2. encouraging thinking processes e.g., McKinney (1996);
3. making individuals able to learn in group settings e.g., Hagan and Sheard (1998);

4. creating motivation and interest e.g., Jenkins (1998);
5. linking new knowledge to what is already known e.g., Doolittle (1998); and
6. making learners responsible and helping them to know what they do not know e.g., Jenkins (1998).

Knowledge construction can be used in Thai university settings to help students in their learning processes. For example, Thai students can construct their own meaningful knowledge since they are engaged to be active learners in constructivist environments. These features can be used to help students develop a better understanding in syntax, semantic, and strategic knowledge in learning to program as they are encouraged to make the learning meaningful by being active learners.

2.3.2 Collaboration

Collaborative learning is an active exchange of ideas which increases both interest among learners and enhances critical thinking through discussion, the clarification of ideas, and the evaluation of others' ideas (Gokhale, 1995). Collaboration is strongly supported as a way to promote knowledge construction. Collaborative learning is essential to encourage individual learners to share and exchange their ideas (Hsi, 1997). It provides opportunities for learners to discover, analyse, synthesise, and evaluate each other's ideas (Gokhale, 1995; Norman & Sphorer, 1996). It is a further learning approach, which shifts the locus of classroom authority from being teacher-centred to being student-centred (Bruffee, 1995; Carlos, 1998). Thus, collaborative learning helps students learn by working together on substantive issues (Bruffee, 1995). As Enerson et al. (1997, para. 1) state "Clearly, collaborative learning is another useful method that can help teachers and students accomplish specific goals". Collaborative learning accommodates a constructivist approach and it is a learning support which Oliver (1999b) claims to be an extremely significant factor in student learning.

Gokhale (1995) has studied the effectiveness of individual learning versus collaborative learning in enhancing critical-thinking skills and drill-and-practice skills. His subjects included forty-eight undergraduate students in industrial technology at Western Illinois University. The results indicated that students who participated in collaborative learning ($M = 12.21$) performed significantly better on critical-thinking than students who

studied individually ($M = 8.63$) whereas on the drill-and-practice both groups did equally well ($F = 3.69$, $p < 0.001$).

A good example of a computer-based collaborative learning system is "The Collaborative World Wide Web Environment Support System (CWEST)" (Oliver, Omari, & Herrington, 1998). This is a WWW-based program that enables university teachers to develop collaborative online learning activities through the provision of customisable templates. The system enables learners to select the mode of collaboration or the cooperative space such as a debate, structured controversy etc. Thus, it encourages learners to become active by contributing to the learning process.

Students who learn individually are limited to information provided in the class itself. They are not able to get alternative explanations or input from other students (Dalton, Hannafin, & Hooper, 1989). Dalton et al. (1989) have studied the students' performance between the use of cooperative and individual computerised instructions using 60 eighth-grade students. The results showed that students who worked cooperatively demonstrate significantly better performance than those who worked individually.

Swigger, Brazile and Shin (1995) have developed a computer-supported cooperative problem-solving environment designed to teach undergraduate students in a computer science major. It provides students with opportunities to work cooperatively and to be active learners and gives teachers the ability to monitor both individual and group performance. Results have shown that students who used the computer-supported cooperative environment performed better than those who did not. This study indicates that group learning provides opportunities to discuss and complete work effectively and efficiently. It is a major point in support of collaborative learning and is supported by Oliver et al. (1996) who have also demonstrated that collaborative activities are advantageous in educational environments.

In collaborative learning, the teacher plays a role as a facilitator for planning, setting up, and running the learning process (Trentin, 1999). Collaborative learning environments invite and encourage students to be active learners who share, discuss, evaluate, discover, analyse, synthesise, and evaluate other's ideas. In teaching programming, it enables students to share the abstract concepts which are necessary for their learning. In collaborative learning students not only work together to complete tasks but also engage

in a process which helps them define and create information that transforms into knowledge through knowledge construction.

The literature suggests therefore, that collaboration can be used to provide a better setting for learners in learning to program through:

1. sharing ideas and learning from other students e.g., Hsi (1997);
2. resolving cognitive conflicts when discussion that leads to new knowledge and better understanding e.g., Norman & Sphorer (1996);
3. motivating and raising interest e.g., Gokhale (1995); and
4. supporting learners and teachers e.g., Enerson et al. (1997).

Collaborative setting can be used to help Thai students to learn computer programming successfully since they have opportunities to collaborate with their peers and discover new knowledge. It can be used as a channel for students to construct their own knowledge when they interact and communicate with peers. It could also help teachers who have to teach computer programming in a large class. These features can be used to help students develop a better understanding of syntax, semantic, and strategic knowledge in learning to program as they have a chance to collaborate with their peers to enhance critical-thinking and drill-and-practice skills.

2.3.3 Visualisation

Visualisation is a learning strategy which uses images, graphics and diagrams to help learners understand abstract concepts. Visualisation has been used to represent abstract business or scientific data as images that can aid in understanding the meaning of the data. It is a powerful problem-solving tool which people use in everyday life to translate and build their understanding (McLoughlin, 1997; Rieber, 1995). For example, when giving directions people visualise the space they are describing. As Rieber (1995) notes "It is interesting how often the direction giver starts with a purely verbal description, but then reverts to visualisation tricks extemporaneously (such as pointing in the air to illustrate the many turns and distances)" (p. 46). The idea of using graphics and animation to illustrate abstract concepts in the learning process such as when learning computer algorithms or mathematical notions was first born in 1981 (Hansen, Narayanan, & Schrimpsher, 2000). Denyer (1997) found that students of Biochemistry had difficulties with chemical calculations and lost their confidence in the learning

process. To address this he designed a computer-based tool for teaching Biochemical calculations by incorporating graphics and animations to help students visualise problems, learn problem-solving strategies effectively, and build their confidence. Each module contained interactive quiz questions and related online help. Results from the use of the learning tool have shown positive outcomes.

To enhance students' understanding of computer programming, Yang (1998) concludes that effective programming instruction must provide graphical representations for students to visualise complicated programming concepts. Warendorf (1997) has developed an intelligent tutoring system for teaching a data structures course called the Animated Data Structure Intelligent Tutoring System (ADIS) which is designed to enhance students' understanding of linked-lists, stacks, queues, trees, and graphs by displaying these structures graphically. The system also includes a tutorial mode where students learn basic algorithms such as insertion, deletion, etc. visually. With this system students spend less time learning to use and manipulate data structures (Warendorf, 1997).

Hansen et al. (2000) have developed an algorithm animation system called Hypermedia Algorithm Visualizations (HalVis) to teach algorithm design at various levels of abstraction. HalVis includes the four typical types of sorting algorithms: BubbleSort, SelectionSort, MergeSort, and QuickSort, and a graphing algorithm. They conducted eight experiments over a period of three years with a total of 232 computer science undergraduate students, with results indicating that the use of HalVis was significantly more effective than traditional teaching methods.

Another sorting algorithm animation system is called Sort Animator (Dershem & Brummund, 1998). Sort Animator is implemented as a Java applet and is accessible through the World Wide Web. It can show both sort animation and code animation synchronously. The sort animation is represented by a row of vertical bars of different sizes to represent different values. As the sort algorithm is executed, the bars move according to the current line of code. The user is able to control the speed of execution, the number of elements being sorted, the colour of the bars, background, and the highlighted line on the code. Another option in this sort algorithm is an explanation window button which displays a text-based description. At the bottom of the screen it displays the number of comparisons and swaps. Dershem and Brummund (1998)

conclude that the visual animation of the Sort Animator provides significant benefits to both students and teachers in understanding of sort algorithms. Its accessibility provides students with the opportunity to learn at their own pace.

In a course on the design and analysis of data structures and algorithms, Goodrich and Tamassia (1998) found that the key concepts were very abstract and not clearly understood by the majority of students because they were all in the form of sophisticated mathematical arguments. Goodrich and Tamassia, therefore, proposed the use of pictures that visualised proofs to enhance students' comprehension. Visual proofs such as summing linear terms, counting nodes in a binary tree, analysing binary tree traversal, etc. were then more effectively learned.

The use of animations for presenting algorithms in computer science discipline can benefit student understanding because students can understand things better when viewed graphically (Bergin et al., 1996). It enables students to see abstract information or algorithms in the form of pictures of what is happening with an explanation of each step in a task (Brummund, 1997; Dershem & Brummund, 1998).

The potential of graphics in teaching and learning environments has led many researchers to optimise its benefits by developing visualisation tools and systems to effectively enhance students' understanding. Further examples of visualisation include, Dershem and Vanderhyde's (1998) Java application that produces a window containing a visualisation for teaching object-oriented concepts and Pierson and Rodger's (1998) Java application called JAWAA which is a Web-based animation program for creating animations of data structures and algorithms, etc.

The literature provides support for using visualisation to enable learners to build a deeper understanding of the ways that programming structures operate and to assist them in understanding the semantics. Thus, we can conclude that the use of animation and graphics images which is very popular in educational environments can be used to support teaching and learning of computer programming. Generally, it can be used to:

1. help to provide concrete examples, e.g., Goodrich and Tamassia (1998);
2. build mental models, e.g., Smith and Webb (1998);
3. help to distinguish between concepts e.g. Bergin et al. (1996); and
4. make it easier to understand concepts, e.g., Jehng and Chan (1998).

Visualisation tools and systems can also be used in Thai university settings since teaching and learning computer programming in Thailand has many problems including:

1. English as a foreign language;
2. large classes and few teachers;
3. busy students who have limited time with trimester system; and
4. students with limited backgrounds.

2.3.4 Expert Performance

Students can learn by observing experts doing the same thing. The learning is enhanced by watching and copying the modelling of processes from an expert. Sometimes learning can occur when students listen to an expert and legitimately acquire information to construct their own knowledge. For example, in the article *Stolen Knowledge* (Brown & Duguid, 1993), the authors begin with a quotation from Bandyopadhyay (1989, p. 45):

> *A very great musician came and stayed in [our] house. He made one big mistake . . . [he] determined to teach me music, and consequently no learning took place. Nevertheless, I did casually pick up from him a certain amount of stolen knowledge.*

This quotation provides such a good example on how learners can learn from an expert and can also "... steal their knowledge from the rich resource made up of other, more experienced workers and ongoing, social shared practiced" (Brown & Duguid, 1993, p. 14).

Expert performances allow learners to observe a task before they actually do it (Herrington, 1997). Novice students can observe the record of an expert's demonstration. This enables learners to absorb strategies used by the expert for that particular task (Collins, Brown, & Newman, 1989). Learners can then compare their understanding or performance to the expert's demonstration at various levels of expertise, an important factor that can support reflection and which leads them to know what elements need to be improved (Collins & Brown, 1988).

Expert performances are one strategy for providing students with information and content from a variety of perspectives. Spiro, Feltovich, Jacobson, and Coulson (1991) contend that "revisiting the same materials, at different times, in rearranged contexts,

for different purposes, and from different conceptual perspectives is essential for attaining the goals of advanced knowledge acquisition (mastery of complexity in understanding and preparation for transfer)" (p. 28). This contention is supported by Young (1993) who describes repeated viewing of the film *Young Sherlock Homes* for an entire semester, suggesting that using the same material for such a long period of time can:

> *invoke images of students bored and tears when viewing the film for tenth or thirteenth time. But learning new perspectives of material that students initially thought they understood completely proved to be challenging and motivating to students. It was the changes in understanding that proved motivating, not the original presentation of the situation. (pp. 49-50)*

The literature therefore suggests that the use of expert performance can be applied into teaching and learning computer programming by providing some tasks and animation examples to be observed by the learner which will help learner to:

1. compare his or her performance (Collins & Brown, 1988);
2. absorb strategies used by an expert (Collins et al., 1989);
3. construct his or her knowledge from the rich resource (Brown & Duguid, 1993); and
4. know what elements needed to be improved (Collins & Brown, 1988).

2.4 What Technology can Support Teaching and Learning Environments

In a traditional teaching setting all information and knowledge were imparted by the teacher; students needed to memorise and repeat what they were given often without a chance to seek meaning and knowledge through experience. Traditional teaching involves the use of blackboards and overhead projections for transferring information which students have limited access to outside the classroom (Schemmel, Hall, & Dennis, 1997). Thus, students are not encouraged to acquire information and knowledge from other sources. However, according to Twigg (1993), "If we anticipate a future where more students need more learning, there is only one way to meet this need without diminishing the quality of their learning experiences: we must change the way we deliver education" (p. 11).

Nowadays, technology has moved into the educational sector providing methods to improve the transfer of information (Schemmel et al., 1997). Many institutions around

the world are taking advantage of new technology, such as the Internet, HyperText Markup Language (HTML), etc. This technology provides an alternative delivery mechanism and can enhance the teaching and learning process.

2.4.1 Technologies and Affordances

In the past, there have been many different media available to deliver information to learners, such as face-to-face instruction, textbooks, overhead projectors, tapes, videos, radio, and television. Each medium has its own unique characteristics and serves a different purpose. For example, a textbook presents information to complement the teacher's presentation, whereas a video replaces the teacher with both static and dynamic images accompanied by sound (Newby, Stepich, Lehman, & Russell, 1996).

2.4.1.1 Computer Technologies

Since the late 60s, computer technologies have been used in educational settings (Oliver, 1999b). The computer is an instructional tool which can be used to transfer text, images, and sound, and to present information or instructional materials (Steinberg, 1991). It is such a powerful teaching tool which not only displays information, but also processes information and its use in education is increasing (Newby et al., 1996). As Dyrli and Kinnaman (1995) state "Whether you currently use computer-based educational technology a lot, occasionally, or not at all, there is little question that your level of use, and that of your students, will increase in the year ahead" (p. 38). Teachers now have the opportunity to use computers and hypermedia authoring software to create interactive instruction and students can produce multimedia reports.

According to Brock (1994, p. 31), computer software in educational settings is divided into 3 categories and can be summarised as follows:

1. Programming software, software used to write programs in computer languages such as BASIC, LOGO, Pascal, or C;
2. Courseware, comprised two types, Computer-Assisted or Computer-Aided Instruction (CAI) and Computer-Managed Instruction (CMI) programs.
    - CAI programs, usually include drill and practice, tutorial simulation, and problem-solving components.

- CMI programs, used to manage instruction, keep records, and to evaluate students' progress.

3. Application software, software used for specific tasks such as word processing, desktop publishing, graphics, database, spreadsheet, or integrated software.

Brock (1994, p. 39) describes some advantages in using programming software and courseware in educational settings in the following table (Table 2.1).

Table 2.1: Advantages of using programming software and courseware

Learners' advantages	Teachers' advantages
• Logic skill development • Problem-solving abilities development • Individualisation of learning tasks • Increasing of motivation • Interactivity with the microcomputer	• Demonstrating and teaching of sequence logic through computer interaction • Saving time with administrative and recordkeeping tasks • Enhancing personalised monitoring of students' performance • Having a computer as a classroom assistance by using CAI and CMI

Vosniadou (1994) notes that technology "...makes it possible to create learning situations that mirror what is happening in the real world in ways that are difficult to realize in a traditional classroom" (p.12). The traditional classroom setting, therefore, seems to be outdated since technology has moved towards educational settings that provide opportunities for information transferring using an interactive approach.

According to a study by Schemmel et al. (1997) on the use of computer technology to improve the transfer of information and student learning:

> Based on student comments, an examination of student work, performance on exams, and final grades some general conclusions can be drawn concerning the effectiveness of these techniques:
>
> - students are focused more on course content than course administration and outline;
> - students have experienced more effective studying;
> - laboratory reports have improved steadily, with fewer errors in lab data;
> - overall, student grades have improved for similar assignments and exams;
> - overall, instructor evaluations have shown steady improvement. (pp. 35-36)

This improvement is supported by a meta-analysis of findings from 254 controlled evaluation studies comparing students learning in traditional classroom settings and technology-based learning (Kulik & Kulik, 1986, 1991) which demonstrated that students using computer-based instruction had more significant achievement than those

who were taught in conventional settings. Kulik and Kulik (1986) also conclude that students who learned the material from computer-based instruction spent an average of 33 percent less time doing so than with traditional teaching methods.

Rais-Rohani (2001) points out that "... a more recent development is the significant shift in technology implementations toward greater use of computers as tools in the learning process rather than as instructional delivery device" (p. 38). He conducted a study of the effectiveness of an online tutorial to enhance static instruction by comparing the students' performance in two different groups, an experimental and a control group. Computer-based instructional materials (CIMS) for the static course were developed for the experimental group to use individually as a self-paced instructional or tutorial tool outside the classroom while the control group did not use the CIMS. The results showed that the experimental group did significantly better than those in control group on test 1, test 2, and the final examination.

This literature suggests that the use of CAI can help learners to:

1. have the ability to control their use e.g., Schemmel et al. (1997);
2. spend less time and gain immediate feedback, e.g., Kulik and Kulik (1986); and
3. be able to use it over and over again, e.g., Rais-Rohani (2001).

The use of computer technology appears to provide opportunities for teaching and learning in computer programming by:

1. facilitating learners to not only perceive information, but also perform their tasks;
2. supporting interactive instruction where learners can produce multimedia reports;
3. increasing learners' motivation; and
4. improving both the transfer of information and student learning.

2.4.1.2 Internet

The Internet is a global communications network connecting million of computers using a standardised set of communication protocols which transfer information among computers (Higgins, 1996). The Internet is a worldwide collection of computer

networks which is accessible to hundreds of millions of people worldwide and enables digital data to be transmitted and received between computer systems (Oliver, 1998).

In 2001, there were estimated 530 million people who used the Internet worldwide and the number of worldwide Internet users by the end of the year 2005 is predicted to double to 1.12 billion (Juliussen, 2002). The Internet is the most widely used communications medium that provides a possible opportunity for teachers and students to meet virtually, to have synchronous communication via chat or asynchronous communication or through e-mail (Tripathi, 1999). Higher education institutions throughout the world are moving toward the new advanced technologies for educational environments by using the Internet as a delivery medium (e.g., online courses) (Corderoy & Lefoe, 1998). The delivery of information using the Internet has more flexibility and has become a preferred alternative method of delivery (Forsyth, 1996; Franklin & Peat, 2000). This delivery method can increase the flexibility in teaching and learning by providing access to a wider range of resources available at any time or place, which teachers and learners can use to maximise their teaching and learning (Reid, 2001).

According to Newby et al. (1996), the use of the Internet can be grouped into three categories based on the applications used: communication, information retrieval and information publishing. These categories can be used to serve the teachers and learners' inquiry as follows:

1. Communication: Communication made through the Internet is either synchronous or asynchronous communication.
    - Synchronous communication: This is communication that takes place at the same time such as Chat, Multi-User Domain (MUD), MUD Object-Oriented (MOO), and Videoconferencing etc. This communication provides an opportunity for teachers and learners who are physically separated by location and need to have live communication.
    - Asynchronous communication: This is communication that takes place over a period of time such as Electronic Mail (e-mail), Usenet newsgroups, Bulletin boards, etc. This communication allows teachers and learners to communicate with each other at their own convenience.

2. Information retrieval: Information can be easily exchanged or retrieved from one computer to another. The primary method for transferring files over the Internet is called file-transfer protocol (FTP) (Oliver, 1998). Telnet is used to create a connection with a remote computer to access an interactive service such as a library catalogue.
3. Information publishing: Information can be published on the Internet by using HyperText Markup Language (HTML) and the World Wide Web (WWW).

Alam and Rencis (1998) believe that active learning can be promoted by using the Internet to develop interactive content such as Web-based discussion, immediate assessment feedback, etc. The Internet can be used to support collaboration, knowledge construction and visualisation in teaching and learning computer programming.

2.4.1.3 The Web

The Web is the most widely used and fastest-growing part of the Internet which can be used for educational purposes by providing content that is easy to distribute, update and is inexpensive and convenient to access. The prominent feature of the Web is the use of HyperText Markup Language (HTML) which is platform independent (Love & Gosper, 1995). The information on the Web has global accessibility. Whalley (1995) demonstrates how the size of the text can be enlarged which enables the students with impaired eyesight to access information more easily. Another feature raised by Alam and Rencis (1998) is that HTML can be extended to produce interactive documents and "...these developments will have a significant impact on the education process" (p. 24). Lemay (1996) and Yang (1998) also point out that the Web is hypertext information system, graphical and easy to navigate, cross-platform, distributed, dynamic, and interactive.

By using the Web, learners can have access to millions of pages of information worldwide by browsing through an application program called a Web browser. Web browsers are available freely on the Internet. The two most typical use of the Web browsers are Microsoft Internet Explorer by Microsoft Corporation (75 percent) and Netscape Navigator by Netscape Communications Corporation (24 percent) (Waller, n.d.). The Web is the most updated resource which provides the latest information for anyone who can connect to the Internet (McIntyre, 1997).

McIntyre (1997) studied a group of 20 adults aged between 27 to 50 who each had different computer skills and Internet experience. The group was asked to rate the Web as a learning tool for adults. Most (17 out of 20) stated that they would continue to use the Web as a research and/or learning tool and those who did not indicate this had little or no previous computer skills or Internet experience. She concluded that the World Wide Web would play a significant role in adult learning since it is a powerful tool for self-directed learning.

Using the Web to store information is low cost and relatively easy to update. This can be useful in an educational setting (Brusilovsky, 2001) where it can be used to support teaching and learning in computer programming. The Web provides:

1. accessibility and flexibility for teachers and students in the acquisition of information at their own pace and time;
2. ease with which to distribute and update;
3. platform independence; and
4. dynamic and interactive information.

2.4.1.4 Multimedia

A frequently cited statistic is that "People generally remember 10 percent of what they read, 20 percent of what they hear, 30 percent of what they see, and 50 percent of what they hear and see" (Treichler, 1967, p. 15). Therefore, as the media used in educational settings comes in many different forms and formats, teachers need to understand the differences between media used in the classroom in order to select the optimum type (Rose & Meyer, 2002). Newby et al. (1996) present 11 types of media and provide the characteristics of each. These are presented in the following table (Table 2.2).

Table 2.2: Media characteristics (Newby et al., 1996, p. 147)

Types of Media	Visual	Color	Sound	Motion	Interaction	Tactile
Real objects and models	●	●				●
Text (books, handouts)	●	●				
Visuals (pictures, photos, drawing, charts, graphs)	●	●				
Display boards (chalk, bulletin, multipurpose)	●	●				
Overhead transparencies	●	●				
Slides and filmstrips	●	●	●			
Audio (tape, disc)			●			
Video and film (tape, disc)	●	●	●	●		
Television (live)	●	●	●	●		

Types of Media	Visual	Color	Sound	Motion	Interaction	Tactile
Computer software	•	•	•	•	•	
Multimedia	•	•	•	•	•	

In addition, Najjar (1996) suggests that information must be put into the most appropriate medium for learning to be most effective. For example, information should be presented as follows:

- Assembly Instructions should be in textual format with supportive pictures;
- Procedural Information should be in the form of explanatory text with diagrams or animation;
- Problem-solving Information should be presented with animation and explanatory verbal narration;
- Recognition and Spatial Information should be presented with pictures;
- Small amounts of verbal information should be presented with sound; and
- Story details should be presented on video with soundtrack or textual format with supportive illustrations.

The term "multimedia" is defined by Tolhurst (1995) as "...the use of two or more media to present information. The media that can be used include text, still or animated graphics, movie segments, sound, and music" (p. 23). These media can be presented by integrating two or more components. Brock (1994) points out that when multimedia is incorporated into instructional design, it encourages learners in self-directed learning and enables access to information from different perspectives. This notion is supported by Bagui (1998) who points out the reasons for an increase in learning with multimedia including interactivity with computers, flexibility, rich content, motivational effects, better structured instruction, immediate feedback, and stimulating style.

Many institutions have adopted multimedia technology into teaching and learning environments as a means of cost reduction and increasing teaching effectiveness (Deacon, Walton, & Wilson, 1997). The most commonly used type of multimedia in educational settings is the interactive multimedia in which teachers and learners can interact (Brown, 1997). Interactive multimedia provides learners with the opportunity to access information in a non-linear fashion in which learners can access information in any of a number of exploration paths. An interactive multimedia production is typically

comprised of five basic types of media including text, sound, animation, video, and graphics which learners can in some way control (Bagui, 1998; Wahle, 1998). This enables learners to monitor and manage their own learning and as a result to gain more motivation for learning.

Brock (1994) indicates that the use of multimedia in educational settings has at least six basic advantages over the traditional classroom. These are that:

- Learning motivation increases through nonlinear interactivity;
- Learning is customized to meet individual needs;
- Resources are multisensory;
- Meta-cognitive abilities (thinking about thinking) plus other higher level thinking skills are encouraged;
- Learning is active; and
- Learning sequences and material selection are more teacher/student-controlled than are in traditional modes of instruction. (p. 195)

An example of multimedia for language learning is Dustin which was developed by the Institute for Learning Sciences at Northwestern University (Schank & Cleary, 1995). Dustin is a multimedia simulation program designed to help students learn English language through learning by doing. The program includes real life situations where students play a role. The program first places the student at O'Hare International Airport where he/she must go through customs, find transportation to a specific place, and check into a hotel. Students interact with simulated people who appear in video clips by typing responses at the keyboard. If the student responds correctly, the program will move to the next task, if not, the program will either break the task into smaller parts or show examples. The program provides the student with control of the learning process by asking what to say, what to do, asking to repeat a message, and asking for a translation. The program also has different levels of difficulty.

Crynes and Hawley (1995) conclude from their review of 139 multimedia programs that there are significant benefits in this mode of learning including greater effectiveness and efficiency with approximately 30 percent less time spent in learning. This also improved student attitudes and decreased the cost of teaching. These findings are supported by a study by Perry and Perry (1998) on university students' attitudes towards multimedia presentations. Perry and Perry's (1998) participants included 84 students enrolled in three computer information systems (CIS) classes at Appalachian State University and 25 students enrolled in a teacher education class at East Tennessee State University.

They experienced 5 weeks of teaching with multimedia presentation as a primary teaching method. Students were asked to indicate their preference among several presentation methods including straight lectures, chalkboards, overhead transparencies, and multimedia presentations. The results were as follows:

- Ninety-seven percent of the students preferred to attend the multimedia presentation class and felt that it was more interesting;
- Ninety-six percent of the students felt it was more enjoyable;
- Ninety-four percent of the students felt that multimedia presentations could hold their attention better;
- Ninety-five percent of the students believed that the instructor could cover more material with multimedia;
- Ninety-two percent of the students thought that they learnt better with multimedia;
- Eighty-five percent of the students agreed that it was easy for them to understand difficult concepts with the use of multimedia; and
- Ninety-three percent of the students thought that they retained course material better when the instructor used multimedia.

A further comparative study is that of Gutwill-Wise (2001) who investigated the impact of active and context-based learning in introductory chemistry courses by comparing students' performance between an experimental group who used an interactive classroom format and a control group who used textbook and lecture format. The study was carried out at two institutions, a small college and a large university. The results showed that the experimental group in the small college outscored the control group in conceptual problems in chemistry and on scientific thinking problems. The same findings were demonstrated at the large university where the experimental group also outscored the control group in the subsequent organic chemistry course.

Haddon, Smith, Brattan, and Smith (1995) conducted a study on whether learning via multimedia could be of benefit to weaker students. Sixteen chemistry undergraduate students were selected and randomly allocated into two groups, an experimental group using multimedia and a control group using a conventional lecture format. Haddon et al.

(1995) used questionnaires, informal interviews, tests, and monitoring of the learning process to:

- evaluate the learning effectiveness between the two groups;
- examine the individual learning patterns of the experimental group;
- assess the attitudes, motivations and criticisms of students; and
- compare academic ability and learning patterns.

The results showed that there was no significant difference between the end of course scores of both groups. Haddon et al. claimed that multimedia was proved to be an effective teaching medium in this setting. A comparison of the time spent using multimedia by each student in the experimental group showed that students who spent the least time on the task performed poorly. Nonetheless, the students using the multimedia were satisfied that it had been an effective and motivating learning method compared to conventional lectures.

The literature suggests that existing technologies are available to improve teaching and learning environments in computer programming courses where technologies are needed to enhance students' understanding in abstract concepts. Some of the technologies that could be adopted include:

1. Computer: CAI provides drill and practice, tutorial, simulation, and tasks which helps develop aspects of syntactic knowledge;
2. The Internet: A medium to transmit and receive data or information which has communications capability to help learners;
3. The Web: An information content that is platform independent, easy to update and low cost, and provides flexibility and accessibility for learners; and
4. Multimedia: This helps students to visualise and make meaning from abstract information and supports visualisation as a learning strategy.

2.4.2 Teaching and Learning Opportunities with Technologies

In this technological and information age, teaching and learning styles using textbooks or chalkboards seem to be an inappropriate strategy, especially in the fields of science and engineering. This mode of teaching supports passive learning. Also instruction on a

blackboard or overhead is usually incompletely transferred to students' notes and students are less able to access instructions outside the classroom (Schemmel et al., 1997). Instructional settings can be designed to optimise learning by using technologies. The Internet and the World Wide Web (WWW) are the most prominent technologies in the educational sector and Kerlin, Kerlin and Obrien (1997) present ten reasons why schools should use the Internet and the World Wide Web:

1. Equity of access to new and evolving forms of literacy;
2. An infinite resource of information;
3. A window to the world;
4. Teachers as learners;
5. Students as learners: active participation;
6. Motivational influence of authentic learning activities;
7. A new mode for self-expression and presentation of self;
8. Community and the role of audience;
9. Student inquiry and cooperative learning; and
10. Assessing and improving student progress.

2.4.2.1 Educational Resources

In 1997, Macromedia announced and promoted the software program Flash for creating graphic content for the World Wide Web (Ulrich, 2001). Flash is a very popular animation tool for Web designers to utilise its features. "As of April 11, 2001, Flash was installed on 88.23 percent of all Netscape browsers, up from 45 percent on Jan. 8, 1999" (WebSideStory, 2001, para. 2). Flash is a special website design tool which comprises all the elements needed to create an interactive website such as graphics, animation, interface elements and interactivity as well as the HTML necessary to display those elements as a Web page on a browser. It is a tool that delivers scalable vector images that are also smaller than bitmapped images. This ensures that the elements created in Flash will be maintained when the viewer resizes the browser window. Flash also provides a streaming capability that allows some elements to be displayed immediately while more information continues to download.

There is also other software that can be incorporated into the instructional design process including:

- Microsoft Word: This is a word processing application from Microsoft corporation. Teachers can use it to write instructional materials while students can use it for taking notes or writing reports;
- Microsoft PowerPoint: This is designed by Microsoft Corporation for creating presentations. It is a high-powered tool used for dynamic presentations in a slide format. Text, images, sound effects, and charts can be embedded into the slide (*Powerpoint in the Classroom: Teacher's guide*, 1998). As Essaka (1998) claims, this software helps teachers or instructional designers to: (a) easily design and plan their presentations with provided templates; (b) create effective multimedia presentations by incorporating text, animation, sound etc.; (c) deliver in different formats such as 35 mm slides, handouts, etc.; and (d) publish on the Web. This tool is useful in creating instructional materials for presentation.
- HyperText Markup Language (HTML): HTML is an authoring language used to create hypertext documents on the World Wide Web. It defines the structure and layout of a Web document by using a variety of tags and directives inserted into a plain text file (Love & Gosper, 1995). The greatest advantage of HTML is that it is platform independent.
- JavaScript: JavaScript is a simple scripting language developed by Netscape Communications Corporation that enables Web authors to design interactive sites. It can be embedded or integrated into an HTML file to create a complex program that interacts with users by obtaining data, processing it and then outputting a return result (Ritchey, 1996).
- Portable Document File (PDF): PDF is developed by Adobe Systems, Inc. This software allows authors to convert any popular documents format to an Adobe Portable Document Format (PDF) file. The PDF file then can be viewed via a Web browser and the 'Adobe Acrobat' plugin which reads the PDF file. Users can also download the read-only program free of charge which is called 'Adobe Acrobat Reader' from http://www.adobe.com. This is a stand-alone program that opens the PDF files without using the Web browser.

The use of the WWW is growing rapidly in educational settings. As Sangster (1995, para. 2) states the "World Wide Web represents a new concept in technology, the library on your desktop, the dictionary at your finger tips, the sound at your ear. There is nothing that we hear or see that will not be accessible through WWW". The WWW can be used to optimise the teaching and learning process. Oliver (1998) suggests four main categories for educational materials and applications on the WWW as in the following table (Table 2.3).

Table 2.3: Instructional form of WWW materials (Oliver, 1998, p. 17)

Form	Description of Materials
Information Access	Convey information alone to the learner, for example a course syllabus, a calendar, assignment, descriptions, lecture notes, workshop descriptions etc.
Interactive Learning	Involve instructional elements that engage the learner, encourage reflection and decision making and provide feedback in response to learners actions
Networked Learning	Provide a means for the organisation, communication and exchange of ideas and information among learners and teachers and other parties in the learning process
Materials Development	When the WWW is used as a means for learners to create and publish materials, the WWW is used as a tool for gathering and collecting information and presenting that information in the published form

2.4.2.2 Web-Based Instruction

There are many definitions of Web-Based Instruction (WBI). For example, Relan and Gillami (1997) define WBI as "...the application of a repertoire of cognitively oriented instructional strategies within a constructivist and collaborative learning environment, utilizing the attributes and resources of the World Wide Web" (p. 43). Khan (1997) defines WBI as "a hypermedia-based instructional program which utilises the attributes and resources of the World Wide Web to create a meaningful learning environment where learning is fostered and supported" (p. 6). From these definitions, we can conclude that the use of WBI can utilise the flexibility of the Internet and the World Wide Web as a transfer medium.

Web-based instruction is often a cheaper delivery system than traditional face-to-face instruction (Brooks, 1997). For example, WBI can be designed to provide synchronous and/or asynchronous communications between teachers and learners regardless of time and place (Yang, 1998). Teachers provide instructions or content to learners by either having live communication or by posting messages to learners. Learners have an opportunity to use a self-paced learning style when accessing their course materials through the Web via an Internet connection at any time of day or night. They can also acquire significant amounts of information from other sources available on the Internet

to fulfil their resource needs. This accessibility can enhance students' knowledge acquisition, their active involvement in the learning process, and their access to immediate feedback (Karuppan, 2001).

Maltby and Whittle (2000) have compared students' perceptions and performance using traditional face-to-face teaching methods using PowerPoint slides and online Web-based delivery methods using computer graphics and hypertext. They used an introductory programming class using the *blackboard courseinfo* shell as a mechanism to deliver the content. Two campuses of Southern Cross University, Lismore and Coffs Harbour campuses, were chosen for the study. The results showed that the online delivery method was feasible and practical, however the majority of students (58 percent) preferred face-to-face lectures and thirty-eight percent did not care. The results also showed that students with high ability had the same achievements in learning in both the face-to-face and online delivery methods.

Another example of the use of WWW technologies to enhance instruction and learning is in a structural materials course in engineering at the Department of Civil Engineering at the University of Arkansas by (Schemmel et al., 1997). Schemmel et al. (1997) used the computer network to create a homepage to pass information and instruction to the students. Students claimed that this had benefits over the traditional style of course presentation including time effectiveness in using classroom and laboratory and increasing students' involvement and performance.

There are a range of development resources including Flash, Microsoft Word, Microsoft PowerPoint, HTML, JavaScript, and Adobe Portable Document Format (PDF) that can be used to enhance teaching and learning environments in computer programming courses. In addition, the Web-based instruction also can be used for computer programming courses as it is cheaper than face-to-face system and provides learners with flexibility in learning.

2.4.3 Instructional Design for Online Settings

Along with the availability of the technology, online learning has been growing rapidly in all educational sectors (MacDonald, Stodel, Farres, Breithaupt, & Gabriel, 2001). However, "... simply publishing a World Wide Web page with links to other digital

resources does not constitute instruction" (Ritchie & Hoffman, 1997, p. 135). Instructional design principles must be used to ensure that the instruction will attract learners' attention, encourage learners to be active, and motivate learners in the learning process.

Madhumita and Kumar (1995) argue that guidelines for instructional design cannot be applied to a particular course or school but apply generally. They present twenty-one guidelines for effective instructional design across disciplines. These follow a sequence of planning, preparation, implementation, and evaluation, as follows:

1. Begin with objectives and keep objectives in focus from planning to evaluation;
2. Establish the initial profile of the learner and his/her expectation before teaching;
3. Match the level of instruction to the learner's level of reception;
4. Motivate the learner by introducing the subject vis-à-vis its future relevance and important;
5. Provide advance organizers to constitute "ideational scaffolding" in learning;
6. Divide a complex task into smaller, achievable learning units and subunits in term of primacy of events;
7. Employ different methods of schematization to promote perceptual organization;
8. Organize complex information in easy-to-remember structures;
9. Associate a new stimulus with a natural response in order to create faster learning;
10. Vary activities during learning in order to sustain learner's attention;
11. Assess the learner's understanding and interact through questions and answers;
12. Allow time for cognitive processing in order to internalize the concepts;
13. Create and provide environment conducive to learning;
14. Ensure the achievement of critical tasks by employing mastery learning techniques;
15. Develop higher mental abilities through participatory teaching;
16. Develop learner's metacognitive skills by employing different strategies;
17. Follow a 'variable-ratio schedule' of reinforcement in order to sustain the interest of the learner;
18. Plan and practice instruction in a variety of ways to match different learning styles;
19. Provide immediate feedback to the learner's responses;
20. Prepare self-learning exercises, for example, assignments, library readings, etc. for the learner; and
21. Conclude the instruction by recapitulating the salient points and by linking with future learning. (pp. 59-60)

MacDonald et al. (2001) point out four features of online learning:

1. Convenient, flexible, and cost-effective means of education that supports a diverse range of learners;
2. Provides learners with access to a wide range of educational resources;
3. Supports an active and dynamic learning environment; and

4. Provides opportunities to interact whereby learners can create a personally meaningful experience. (p. 15)

Oliver and Herrington (2001) propose the more common factors of online delivery including flexibility, economy, and enhanced learning. However, before developing instructional materials into an Internet delivery mechanism, Forsyth (1996) suggests a Strengths, Weaknesses, Opportunities, and Threats (SWOT) analysis, as outlined in the following table (Table 2.4).

Table 2.4: An example of SWOT analysis (Forsyth, 1996, p. 37)

Strengths	**Weaknesses**
Internet a growth area: increases exposure for course offerings Course material already instructionally designed Industries moving into Internet-capable operations opens up potential for learners to access course material on the job Use human ability to control and program computers	Currently access to the Internet for some learners Course material already instructionally designed (but not revised for the Internet) Reluctance of teachers to use technology • Lack of an Internet standard • Possibility of electronic page-turning if the course materials lack design
Opportunities	**Threats**
Frees up teaching spaces Opens up enrolment opportunities Recognition as education and training leader if best practice followed Reuse existing resources, giving cost benefit	To status as provider if not at the leading edge To teachers who need to alter their teaching style To credibility if appropriate implementation

2.4.3.1 Constructivist Learning Settings

Numerous writers have provided guidance for the design of constructivist learning settings. Some have provided the beginnings of a framework for a constructivist approach. For example, Lebow (1993) presents five principles that support constructivist learning:

1. Maintain a buffer between the learners and the potentially damaging effects of instructional practices.
2. Provide a context for learning that supports both autonomy and relatedness.
3. Embed the reasons for learning into the learning activity itself.
4. Support self-regulation through the promotion of skills and attitudes that enable the learner to assume increasing responsibility for the developmental restructuring process.
5. Strengthen the learner's tendency to engage in intentional learning processes, especially by encouraging the strategic exploration of errors. (p. 5)

Cunningham, Duffy and Knuth (1993, pp. 21-29) suggest seven constructivist principles as a framework for building constructivist learning environments:

1. Provide students with experience with the knowledge construction process;
2. Provide experience in and appreciation for multiple perspectives;
3. Embed learning in realistic and relevant contexts;
4. Encourage ownership and voice in the learning process;
5. Embed learning in social experience;
6. Encourage the use of multiple modes of representation; and
7. Encourage self-awareness of the knowledge construction process.

These seven principles were put into practice by Honebein (1996) who examined two constructivist learning environments, the Lab Design Project (LDP) and the SOCRATES curriculum (Student-Oriented Curriculum: Reflection and Technology as Educational Strategies). He suggests that "... these goals provide just the framework; the designer's interpretation of the goals and subsequent translation into learning activities is the real art in the design of constructivist learning environments" (p. 18).

Savery and Duffy (1995) characterise the philosophical view of constructivism in terms of three propositions:

1. That understanding is in our interactions with the environment.
2. That cognitive conflict or puzzlement is the stimulus for learning and determines the organisation and nature of what is learned.
3. That knowledge evolves through social negotiation and through the evaluation of the viability of individual understandings.

From these constructivist propositions, Savery and Duffy (1995) derive the following set of eight instructional principles for the design of a constructivist learning environment in a problem solving context. They also believe these to be central to the principles of learning:

1. Anchor all learning activities to a larger task or problem.
2. Support the learner in developing ownership for the overall problem or task.
3. Design an authentic task.
4. Design the task and the learning environment to reflect the complexity of the environment they should be able to function in at the end of learning.
5. Give the learner ownership of the process used to develop a solution.
6. Design the learning environment to support and challenge the learner's thinking.
7. Encourage testing ideas against alternative views and alternative contexts.
8. Provide opportunity for and support reflection on both the content learned and the learning process. (pp. 32-34)

Albion (2000) presents nine principles for the constructivist educational design of interactive multimedia and problem-based learning (IMM-PBL) for materials which can be accessed by a web browser or delivered on a CD-ROM. The nine principles are as follows:

1. Begin with an authentic problem;
2. Incorporate relevant cases;
3. Represent multiple viewpoints;
4. Stimulate the activation and elaboration of knowledge;
5. Scaffold learner performance;
6. Provide a strong narrative line;
7. Provide access to relevant information;
8. Encourage self-evaluation; and
9. Support individual and collaborative learning.

Oliver (2000) points out that there is little information and few guidelines for instructional designers to support constructivism. He suggests 6 guidelines to be used to support a constructivist learning environment as follows:

1. Choose meaningful contexts for learning;
2. Choose the learning activities ahead of the content;
3. Choose open-ended and ill-structured tasks;
4. Make the resources plentiful;
5. Provide supports for the learning; and
6. Use authentic assessment activities.

Oliver and Herrington (2001) argue that there are three interconnecting elements which are critical components for the design of constructivist learning settings:

1. Learning tasks;
2. Learning resources; and
3. Learning supports.

Oliver and Herrington (2001) state that "In the design process, it is possible to include and omit various elements. However, in the design of effective constructivist settings, it is important to include particular elements" (p. 20). They provide some forms of learning design that can be used for the learning tasks including problems, investigations, inquiries, projects, and role plays. Some supports and resource based

activities need to be provided for successful outcomes. These supports are in the form of tutorials, quizzes, simulations, worksheets, teamwork, collaboration, conferences, or mentors.

Oliver and Herrington (2001) also provide some examples of relevant learning resources including books, databases, papers, documents, articles, notes, manuals, references, and web links. In addition, learning supports include course schedules, instructions for students and procedural descriptions, and announcements and messages given by the instructors. The following figure (Figure 2.1) shows the various components of a framework for designing online learning settings that support knowledge construction.

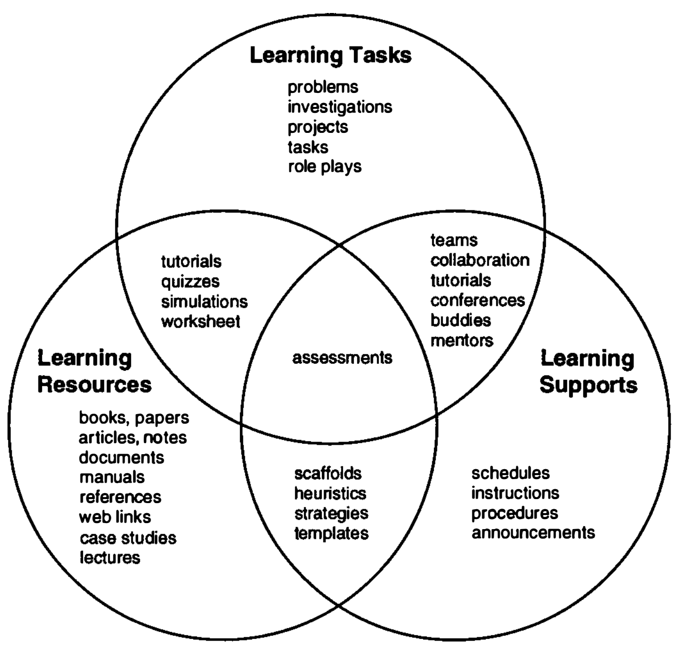


Figure 2.1: Constituent elements of online learning settings (Oliver & Herrington, 2001, p. 20)

This suggests that the design of an effective constructivist setting is needed to incorporate three major elements: (a) learning tasks, (b) learning resources, and (c) learning supports. A computer-programming course can be designed by including problems, tasks, references, web links, lectures, books, schedules, instructions, procedures, announcements, simulation, worksheets, collaboration, tutorials, quizzes, and assessments.

2.4.4 Instructional Design for Programming

Hagan and Lowder (1996) have designed instructional materials for teaching introductory computer programming in the C++ language. Their main home page contains (a) information about the course, and (b) links to connect to 8 different topics which include:

1. Announcements: Relevant course information for students such as sample test plan, related textbooks, staff homepage and e-mail addresses;
2. Assignments: Information relating to assignment specification;
3. C++ Information: Providing code examples, demonstration of ideas and concepts for help in the programming aspects;
4. Exercises: Hints and solutions for programming exercises are uploaded onto the Web;
5. Feedback: Providing space for students to post their comments which are sent to staff via e-mail;
6. Lectures: Collection of PowerPoint slides taught in class;
7. Newsgroup: Providing topics for student to discuss relating to programming problems; and
8. Tutorial Pages: Allowing each tutorial group to post their presentations on the Web.

Hagan and Lowder (1996) also present guidelines on creating online course material on the WWW including Interface design and structure, Download time, Feedback, and Information retrieval. They conclude that students prefer to have more support from the staff and information on the Web, accessibility from home, and collaborative learning by using the Newsgroup and Tutorial Pages.

Yang (1998) states that the two prominent features of the WWW include multimedia and non-linear accessibility and has developed a Web-based learning system for teaching an undergraduate programming class in C programming at the University of Nebraska-Lincoln. He designed the learning system with HTML, JavaScript and CGI and then explored the feasibility and effectiveness of the instructional setting via the WWW by conducting two studies, Spring sessions with 27 students as a control group and Summer sessions with 20 students as an experimental group. Both groups had the

same traditional classroom instruction but the experimental group also had the Web-based learning system.

Yang's (1998) Web-based learning system was comprised of four main features: (a) Tutorial Course—lecture assignments, quizzes, and examinations; (b) Discussion Board—classroom discussion; (c) Ask Questions—space for question submission; and (d) Online References—further reading. Results showed that there was no statistically significant difference in the two groups' achievement. The majority of students agreed that the Web-based instruction was more interesting, gave them more confidence, and improved their communication, learning ability and overall learning quality. Also they felt that access to the materials was easy. However, the majority of students disagreed that Web-based learning environment could replace the traditional classroom instruction or could increase their participation.

Naps and Bressler (1998) developed a Web-based visualisation algorithm called "WebGAIGS" which is based on the GRAIGS AV system, a previous system they had developed in 1996. This uses the WWW and combined mixed media such as text, images and sound. The user enters input data to the algorithm in an HTML form. The system then provides a multi-windowed viewing environment to enable users to view several successive states of the algorithm on the same screen. It provides different algorithms such as a quick sort, shell sort, heap sort, insertion into binary search trees and heaps, etc. Naps and Bressler conclude that animations or images have the potential to enhance students' understanding of difficult and abstract concepts. This is supported by Astrachan and Rodger (1998) who note that the use of animation and visualisation can enhance students' comprehension.

Brusilovsky (2001) points out that in the traditional style of teaching computer programming or related courses, experienced teachers use problem-solving examples to demonstrate and explain concepts. However, this approach does not work with a large group of students who have different levels of ability in the learning and acquisition of programming concepts. Brusilovsky (2001) has developed an example-based programming approach on the Web called WebEx, to enable students with different levels of ability to explore the examples with explanations at their own pace. He claims that the static appearance of the explanation of each line in the textbook can distract students from concentrating on the particular point that needs explanation. WebEx is

designed to overcome this problem by showing an explanation on the right side of the source code when students request it (see Figure 2.1).

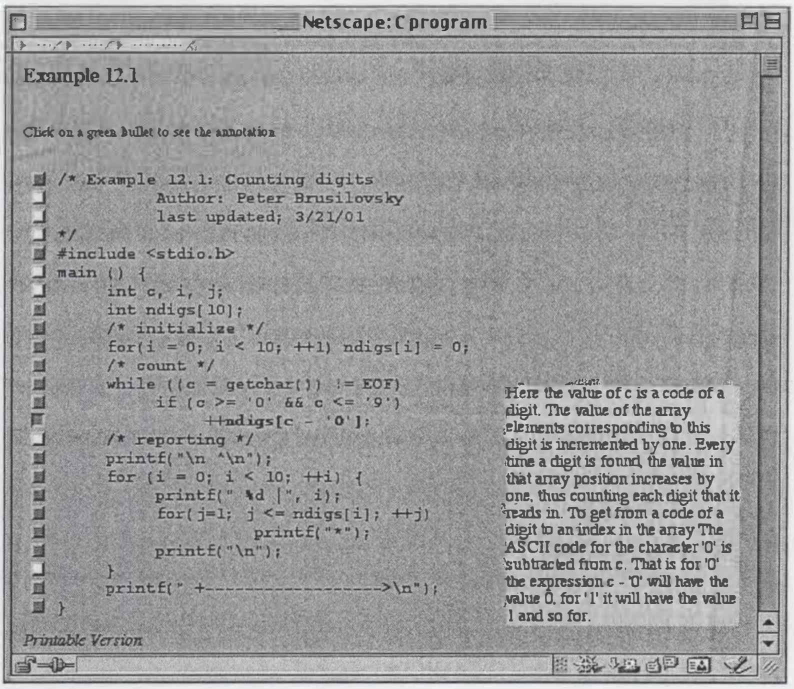


Figure 2.2: A sample of WebEx shot screen (Brusilovsky, 2001).

Brusilovsky (2001) concludes that WebEx has benefits over conventional teaching because it:

1. provides the possibility for students to explore and reuse self-explanatory programming examples;
2. saves time for teachers in the explanation of source code;
3. enables less experienced teachers to teach courses more effectively; and
4. can be used the WebEx database to serve as a community resource.

To this end, Oliver (1999a) points out that learning theories have always required more active learning processes. In order to promote active learning, Holzer (1994) recommends providing opportunities for students to inquire, explore, and collaborate. Interactive multimedia and communication technologies provide these opportunities by enabling collaboration and visualisation in a constructivist learning environment.

2.4.5 Summary of the Literature

The review of the literature conducted in this chapter included:

- the problems and difficulties in students' learning of computer programming;
- strategies teachers use for teaching programming;
- how students learn to program; and
- what technology can support the teaching and learning environment.

This review has provided sufficient information for this research to be conducted by developing an alternative teaching and learning model, based on constructivist learning principles and the instructional design suggested by the literature. The instruction is planned to design as a Web-based application using the Internet as a delivery medium.

The essential elements needed for computer programming are:

1. Student centred;
2. Interaction;
3. Linking syntax and semantics; and
4. Strong visual orientation.

This form of learning setting needs to be designed around a framework comprising learning tasks, learning supports, and learning resources. This framework can be used to plan and develop such a model that can be used to encourage and stimulate students to be active learners and student centred. In the next chapter, Chapter 3: Conceptual Framework, these elements are discussed in more detail and a framework for this study is proposed.

CHAPTER 3: CONCEPTUAL FRAMEWORK

The use of Interactive Multimedia (IMM) via the World Wide Web in educational sectors is rapidly growing (Fetherston, 1998). Fetherson proposes that "Before IMM can be used to best advantage in education, research needs to be conducted that will generate pedagogical guidelines for its use in the various educational contexts" (p. 99).

In this study, the conceptual framework used is based on the learning principles necessary for the successful learning of programming; it is derived from the literature that has been used to inform and guide the design of learning environments. Methods of visualisation, collaboration, constructivism, and student-centred learning are used to explore how the design can benefit teaching and learning in introductory computer programming.

A constructivist learning environment has been adopted in this conceptual framework. The conceptual framework develops, implements, and evaluates an interactive instructional model based on constructivist learning theory and relevant research findings.

3.1 Conceptual Framework for DIVTIC

A possible solution for the conceptual framework comprised of three major parts: (a) form, (b) content, and (c) learning strategy led to the model for the Dynamic Interactive Visualisation Tool in Teaching C (DIVTIC) which is a multimedia-based learning resource to support programming in C among novice learners.

3.1.1 Form of DIVTIC

The form of DIVTIC was planned below:

1. **Web-based instruction:** Web-based instruction has an advantage over the traditional face-to-face instruction because it is easier to update, more accessible, more flexible, and less costly (Brooks, 1997). Thus, DIVTIC was planned to use Web-based instruction as a form to deliver course materials via the Internet.

2. **Flash:** It was planned to use Flash as the major software to develop DIVTIC as follows:
   - create an interactive website including graphics, animation, HTML, and interface elements;
   - enable interactivity between user and the program;
   - provide a streaming capability where large files need to be downloaded;
   - display a step-by-step animation of the programming process; and
   - produce instructional materials and be delivered and worldwide used via the Internet.
3. **Modular:** DIVTIC was planned to include10 modules based on the course material at SUT, the university where the tool was to be evaluated. Students could access any module by selecting it. This feature would accommodate different students' abilities and let them choose their own level of difficulty.

3.1.2 Content of DIVTIC

The content of DIVTIC was planned to comprise six components:

1. **Algorithm-based in C language:** The content was planned by using C language based on the requirement at SUT. DIVTIC was planned to provide students with visual representations of all the major algorithms in the course and for each algorithm that would:
   - show the computational process phase by phase;
   - show the memory mapping, input process, output process, and decision making process;
   - provide explanations of each step; and
   - provide the students with ability to control the process, to pause, go back, go forward.
2. **Virtual computer:** DIVTIC was planned to include as a virtual computer which could display an imitation of a computer monitor for displaying an output and an imitation of the CPU for displaying how the variables and their values were kept in the CPU when students run an animation;

3. **Syntax presentation:** DIVTIC was also planned to provide a dialogue box to display each animation source code with a marker that ran through all segments of each line. This feature would allow students to look at the syntax, variable or keyword, and its explanation;
4. **Explanations:** Some explanations were planned to incorporate with each animation a display of the meaning of each syntax, variable, or keyword when the marker ran past at any particular stage;
5. **Examples:** Each module was planned to include three to four examples ranging from easy to difficult levels. This would benefit students with different learning abilities and motivate students with higher abilities to try more difficult problems; and
6. **References:** DIVTIC was planned to include references as supplementary resource for students.

The design of DIVTIC was drawn from the literature and contained critical elements to support learning in this complex domain.

3.1.3 Learning Strategies Embedded in DIVTIC

The DIVTIC system planned to employ five learning strategies:

1. **Interactive/Feedback:** DIVTIC was planned to provide interactive and immediate feedback which would encourage students to be active learners;
2. **Visual Representation:** DIVTIC was planned to use visual representation to help students understand programming concepts better by visualising what is happening at each stage of the programming process;
3. **Forward/Backward Control:** DIVTIC was planned to include a control menu for students to use while they were watching the animation. This control menu would work in the same way as a video controller and was comprised of *Play, Step-Backward, Step-Forward, Stop/Pause, Go to the End,* and *Go to the Beginning* buttons. This feature would allow students the ability to control the animation process. It also would enable students to pause and think before watching a further step of the animation and this would provide an opportunity for students to become active learners.

4. **Learning from a Computer:** Once the DIVTIC system was installed onto the student's own computer hard disk, students could run the animation section regardless of the Internet connection.
5. **Supporting Normal Classroom Learning:** DIVTIC was planned to contain with all necessary elements needed to support normal classroom activity. But students could also use the DIVTIC system outside classroom at any time of the day, at their own pace.

DIVTIC was intended as a Web-based tool to supplement traditional face-to-face instruction. It would use HTML, JavaScript and Flash. It would contain all the needs of traditional instructional approach. But its features could also be extended to serve as a Web-based course of instruction. The design was intended to create opportunities for student-centred learning and active engagement in a constructivist setting.

3.2 The Planned Functionality of DIVTIC

The design of DIVTIC was conceived to use the Internet as a delivery medium. All relevant course materials were produced and uploaded onto a server. Students were given a unique password to log into the DIVTIC system. A cgi script was used to record the students' log in time including time spent and visited pages for further evaluation.

To ensure that the students' use of DIVTIC was consistent, a weekly task was designed. Students were to be given a weekly task to complete in the laboratory which would require them to log into the DIVTIC system to explore how a program would run and to produce its output. Students would do this by selecting and playing the relevant animation for the task.

DIVTIC was planned to contain eight sections:

1. Computer Structure;
2. Syllabus/Lecture Notes;
3. Animated Examples;
4. C Compiler;
5. C References & Link;
6. C WebBoard;
7. Self-Evaluation; and

8. FAQ Pool.

A description of each planned section is described in the following pages.

3.2.1 Computer Structure

The Computer structure section was planned as an animation which would explain each part of the computer to give students an overview of the basic structure of a computer and to provide the opportunity for students to be familiar with the overall functioning of a computer.

3.2.2 Syllabus/Lecture Notes

This was planned to be a set of course materials and relevant information. To assist in the knowledge construction process it was planned to allow students to manage their own time and construct their own knowledge.

3.2.3 Animated Examples

This was planned to be a set of animation examples which students could interact with by clicking on the control buttons. The animations would show students each step of program execution. A marker would be used to animate each line throughout all the segments of each line of the program. Animation examples would be divided into three different levels of difficulty: (a) easy and short animations, (b) average difficulty animations, and (c) long and complex animations. The dynamic illustration of DIVTIC was planned to conform to Rowe and Thorburn's (1999) contention that illustrations should be made clear to students to assist them to feel comfortable about writing programs.

It was planned that each animation would have four panels used to synchronously display relevant information at any specific time. The design considerations of each animation were:

- short algorithms;
- students viewing more;
- stopping to short action;
- screen with 4 panels;

8. FAQ Pool.

A description of each planned section is described in the following pages.

3.2.1 Computer Structure

The Computer structure section was planned as an animation which would explain each part of the computer to give students an overview of the basic structure of a computer and to provide the opportunity for students to be familiar with the overall functioning of a computer.

3.2.2 Syllabus/Lecture Notes

This was planned to be a set of course materials and relevant information. To assist in the knowledge construction process it was planned to allow students to manage their own time and construct their own knowledge.

3.2.3 Animated Examples

This was planned to be a set of animation examples which students could interact with by clicking on the control buttons. The animations would show students each step of program execution. A marker would be used to animate each line throughout all the segments of each line of the program. Animation examples would be divided into three different levels of difficulty: (a) easy and short animations, (b) average difficulty animations, and (c) long and complex animations. The dynamic illustration of DIVTIC was planned to conform to Rowe and Thorburn's (1999) contention that illustrations should be made clear to students to assist them to feel comfortable about writing programs.

It was planned that each animation would have four panels used to synchronously display relevant information at any specific time. The design considerations of each animation were:

- short algorithms;
- students viewing more;
- stopping to short action;
- screen with 4 panels;

- movement to attract students; and
- visualising programming process.

The plan for each panel was described as follows:

- **Source Code:** A Source Code panel would display a given problem with a button to link to its solution in C source code format. This feature would encourage students to pause, think and solve the problem. Clicking on a button would allow them to compare a given result. This would encourage students to be active learners.
- **Dynamic Explanation:** A Dynamic Explanation would provide some explanations on the behavior of the marker. This was aimed at helping students to develop a valid mental model, which was a critical for novices.
- **Simultaneous Dynamic Memory Map:** A Simultaneous Dynamic Memory Map panel would represent a memory map of the computer. This dynamic memory map was intended to help students develop mental models of how the computer stores data/variables and their values. When the marker ran through the declaration section in the C source code, the equivalent section in the memory map would be highlighted. The number of boxes, which depends on the particular type of variable, would be assigned to that variable individually. For example, a *character, an integer, and a float* type would be assigned 1, 2, and 4 boxes respectively, since they need 1, 2, and 4 bytes to hold their values. Their values would be displayed and changed in the memory location according to the process. This dynamic visualisation was aimed at enhancing students' understanding of how memory has been allocated and changed. The use of this feature is supported by Mayer (1976) and Smith and Webb (1998).
- **Synchronise Output Screen:** A Synchronise Output Screen panel would represent a virtual device used to display the output at each point where the marker was placed. To make it realistic, this screen was planned to design to sit on top of the Simultaneous Dynamic Memory Map panel as shown in Figure 3.1.

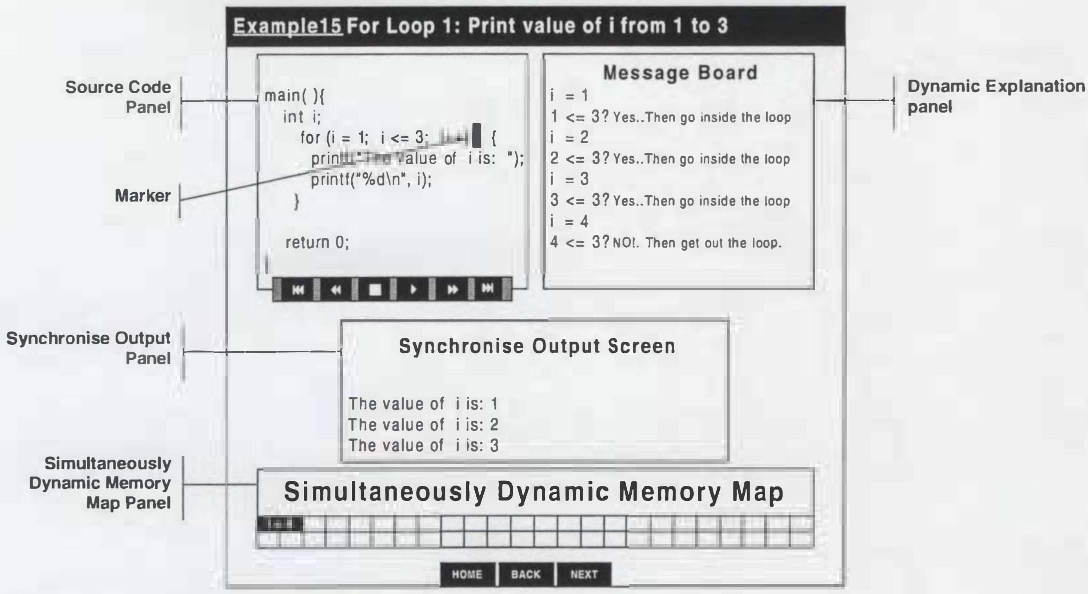


Figure 3.1: A planned layout of an animation showing the four elements

3.2.4 C Compiler

DIVTIC was planned to include a C compiler. This was planned to be a step-by-step animation that would demonstrate how to use a C compiler. It was intended to help students become familiar with the C compiler and also to encourage them to write a simple program. This feature was influenced by Azemi (1995) who argues that the most important thing for novice programmers is practice in writing programs.

3.2.5 C References & Links

In the plan, this would be a kind of information pool, which would assist students in constructing their own knowledge by searching for relevant references on the server and the World Wide Web. This feature would seek to encourage students to be active learners by providing relevant links to start searching for information.

3.2.6 C WebBoard

The literature describing how students learn suggested the value of collaboration. Thus, DIVTIC was planned to include such supporting elements. The C WebBoard was planned to provide opportunities for students to communicate with their peers. This feature was planned to encourage individuals to share and change their ideas (Hsi, 1997) leading to an opportunity to discover, analyse, synthesise, and evaluate each other's

thoughts (Gokhale, 1995; Norman & Sphorer, 1996). It was intended that students could post their questions and receive answers via the use of this feature. Moreover, it would enable students to play a role in a collaborative learning activity which can be a significant factor contributing to student learning (Oliver, 1999b).

3.2.7 Self-Evaluation

The literature describing how students learn also suggested the value of reflection and self-evaluation. For this reason DIVTIC was planned to include a self-evaluation component. This element was planned to allow students to test their understanding. It was planned to be comprised of a set of multiple-choice questions which would cover all topics. It was also planned to provide a dynamic feedback window for students when they clicked on an answer. This option was intended to increase students' motivation to test their understanding of each topic and to provide them with dynamic feedback while using DIVTIC. It would also challenge students to participate and improve their learning outcomes.

3.2.8 FAQ Pool

This was planned to be a knowledge-based pool that contains frequently asked questions (FAQs). This feature was intended to provide students with easy access to some common questions which peers have asked together with their answers. This was planned to be the first place for students to go when they had a question.

Table 3.1 shows the various elements planned for DIVTIC and lists the forms of learning activity each was intended to support together with a list of contributions the activities could provide for students' knowledge construction and learning.

Table 3.1: The relationship between DIVTIC characteristics, forms of learning activity, and its contributions

DIVTIC Characteristic	Intended Student Use	Contribution to Learning
Syllabus/Lecture Notes: *This was planned to be a set of course materials and relevant information. To assist in the knowledge construction process it was planned to allow students to manage their own time and construct their own knowledge.*	• Browsing, exploring, or inquiring when needed during trimester • Preparing for lecture by downloading and printing notes ahead of time	• Support of self-regulated activity • Support for learner inquiry
Computer Structure: *This was planned to be a set of animations that would explain each part of the computer to give students an overview of the basic structure of a computer and to provide the opportunity for students to be familiar with the overall functioning of a computer.*	• Browsing, exploring, or inquiring when needed during trimester • Seeking information	• Support of self-regulated activity • Support for learner inquiry • Support for knowledge acquisition
Animated Examples: *This was planned to be a set of animation examples which students could interact with by clicking on the control buttons at anytime. The animations would show students each step of program execution. A marker would be used to animate each line throughout all the segments of each lines of the program.*	• Browsing, exploring, or inquiring when needed during trimester • Interacting with the animation process • Observing the code at run time	• Support of self-regulated activity • Provision of feedback • Support for knowledge acquisition • Support for higher-order thinking • Provision of multiple perspectives • Scaffold for learning • Support for learner inquiry • Support for learner exploration • Modelling of expert performance
C Compiler: *A step-by-step animation that would demonstrate how to use a C compiler.*	• Browsing, exploring, or inquiring when needed during trimester • Testing source code	• Support of self-regulated activity • Support for learner inquiry • Support for learner exploration • Provision of multiple perspectives • Support for knowledge acquisition • Modelling of expert performance
C WebBoard: *This was planned to provide opportunities for students to communicate with their peers. This feature was planned to encourage individuals to share and change their ideas leading to an opportunity to discover, analyse, synthesise, and evaluate each other's thoughts. It was intended that students could post their questions and receive answers via the use of this feature.*	• Browsing, exploring, or inquiring when needed during trimester • Seeking information • Supporting peers	• Provision of collaborative opportunities • Provision of feedback • Provision of multiple perspectives • Support for knowledge sharing • Articulation of idea
Self-Evaluation: *It was planned to comprise of a set of multiple-choice questions which would cover all topics. It was also planned to provide a dynamic feedback window for students when they clicked on an answer.*	• Testing understanding when needed • Knowledge acquisition	• Support of self-regulated activity • Provision of feedback • Scaffold for learning • Support for learner exploration • Articulation of idea
FAQ Pool: *This was planned to be a knowledge-based pool that contains frequently asked questions (FAQs). This feature was intended to provide students with easy access to some common questions which peers have asked and their answers.*	• Finding common frequently asked questions • Seeking responses to problems	• Support for learner exploration • Support for learner inquiry • Expansion of knowledge base • Scaffold for learning • Modelling of expert performance
C References & Links: *This would be a kind of information pool, which would assist students in constructing their own knowledge by searching for relevant references on the server and the World Wide Web.*	• Browsing, exploring, or inquiring when needed during trimester • Investigating syntax and algorithms	• Expansion of knowledge base • Provision of multiple perspectives • Support for learner inquiry

Figure 3.2 shows the planned overall structure of DIVTIC and demonstrates the linkages and connections between the various elements.

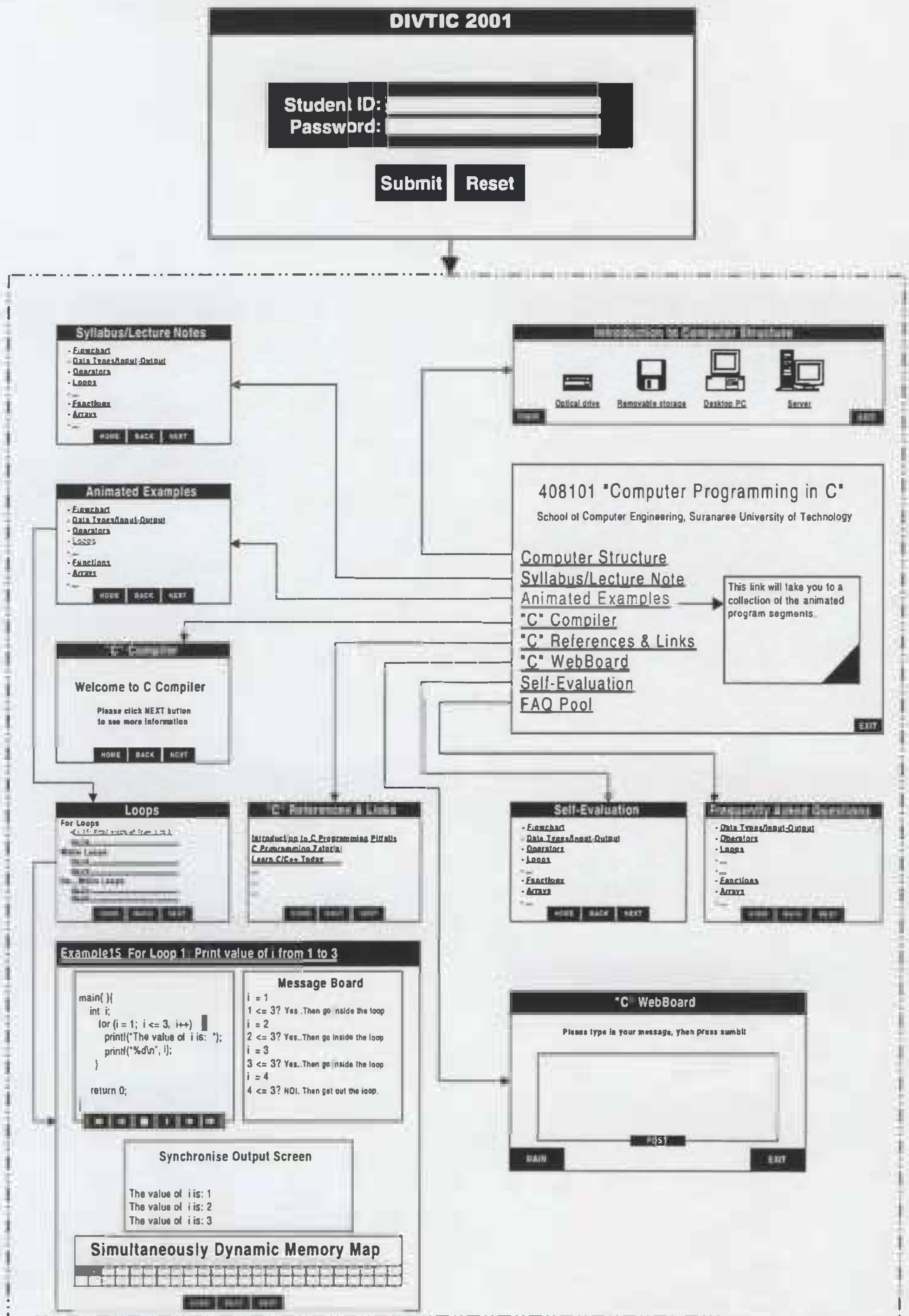


Figure 3.2: DIVTIC's planned structure

The design and characteristics of DIVTIC were intended to address many of the problems in the teaching and learning of introductory computer programming. DIVTIC was planned to:

- encourage students to be active learners by, for example, (a) giving a problem first to pause for the students to think of a solution, then students can click on a button to see the solution, (b) providing more accessibility for students to access the DIVTIC system at any time;
- provide a Source Code panel to help students understand the syntax of the C language with a marker by looking at the meaning in the Dynamic Explanation panel when the marker runs past any line of syntax, variable or keyword;
- provide some explanation in a Dynamic Explanation panel to help students understand the action of each element in the Source Code panel when the marker runs past any line of syntax, variable or keyword;
- help students to visualise each phase of the programming process by incorporating 4 panels: Source Code, Dynamic Explanation, Simultaneous Dynamic Memory Map, and Synchronise Output Screen panels. These would ensure that students could see the change of process inside the computer and lead to a better conceptual understanding of how to construct syntax and semantics to solve a given problem. It would also enhance their mental models of programming to provide a better understanding of program execution;
- save time for teachers in re-teaching the concepts. Students could simply run a relevant animation to improve their understanding at any time;
- provide a channel for students to collaborate with their peers over the Internet. This would encourage students to contribute to the learning process and to engage in a process to define and create information that transforms into knowledge; and
- provide immediate feedback in a Self-evaluation section that would encourage the learning process.

3.3 Research Aims

With DIVTIC planned, the aim of this study was to explore how students in introductory programming courses would be aided by the use of such interactive visual and Web-based instructional materials. This was achieved through the subsequent development of DIVTIC and its implementation in a tertiary setting. A study was planned to explore the following research questions:

1. How do students use the DIVTIC?
   - (1a) Which components of DIVTIC do students use and for how long?
   - (1b) What strategies do students use with DIVTIC?
   - (1c) What factors influence students' use of DIVTIC?
   - (1d) What attitudes do students generate towards DIVTIC?
2. To what extent does the dynamic interactive visualisation process implemented in DIVTIC influence learning outcomes?
   - (2a) How does the dynamic interactive visualisation process implemented in DIVTIC influence students' performance in programming?
   - (2b) How does use of the dynamic interactive visualisation process implemented in DIVTIC vary among students?
   - (2c) What levels and forms of cognitive engagement are evident among DIVTIC users?
   - (2d) What factors influence students' achievement with DIVTIC?

CHAPTER 4: DEVELOPMENT OF DIVTIC

The review of literature suggests that one possible solution to enhance student understanding and learning outcomes is to incorporate multimedia technologies into the learning process. This may include the Internet, authoring tools, online discussion boards, knowledge pools, self-assessment tasks, and visual interface design.

4.1 Development of DIVTIC

For this study, the above programs mentioned in Section 2.4.2.1 have been incorporated into the development of DIVTIC including Flash, HTML, JavaScript, and Acrobat. These were all necessary in producing the final version of DIVTIC for delivery via the Internet so that a specific group could access the system by providing a unique password to each student for logging into the DIVTIC system.

The development of DIVTIC was based on contemporary learning theories, applied using available communication technologies. It was to be more visually explicit than the existing systems (e.g., Rowe & Thorburn, 1999) because DIVTIC shows phrases in each line. The DIVTIC system was also designed to employ a combination of complementary tools that encourage students to be active learners by utilising the Internet as a delivery medium.

The Animated Examples section was the most important section in developing the DIVTIC system. This section contained all the animated examples needed for the entire course. It was divided into 10 chapters which were associated with the course outline as follows:

1. Flowchart;
2. Data Types / Input & Output;
3. Operators;
4. Control;
5. Functions;
6. Arrays;
7. Pointers;

8. Sorting & Searching;
9. Structures; and,
10. Data Files.

Each chapter was comprised of three to six animations depending on how many were needed for the chapter. There were 46 animated examples in total. It took about six months to build. In addition to the Animated Examples section, there were also seven combinations of complementary tools which comprised the learning resources described as follows:

1. Syllabus & Lecture Notes: Self regulation;
2. Computer Structure: Self regulation;
3. Animated Examples: Self regulation and modelling of expert performance;
4. C Compiler: Self regulation and modelling of expert performance;
5. C WebBoard: Provision of multiple perspectives and articulation of idea;
6. Self-evaluation: Self regulation and articulation of idea
7. FAQ Pool: Scaffold for learning and modelling of expert performance; and,
8. C References and Link: Provision of multiple perspectives.

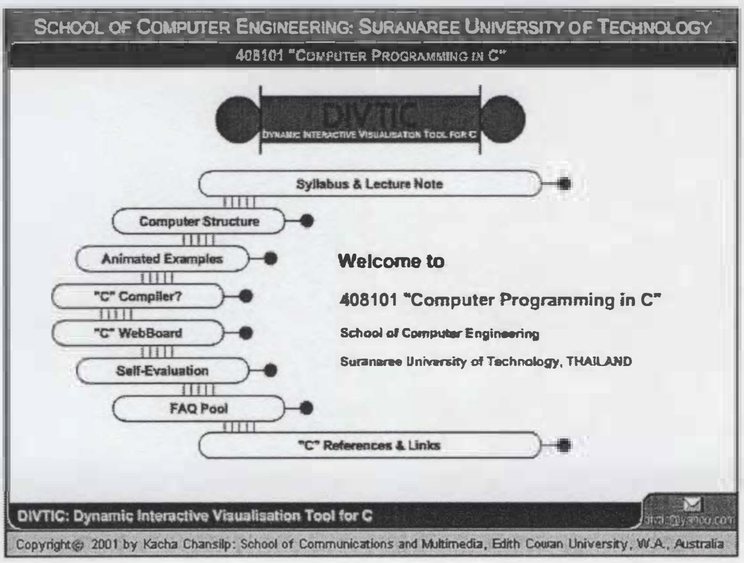


Figure 4.1: DIVTIC's homepage

Each different section in DIVTIC's Home Page (Figure 4.1) with its own unique advantage and usability, are described below in their order of layout from top to bottom as they appear on the first screen.

4.1.1 Syllabus & Lecture Notes

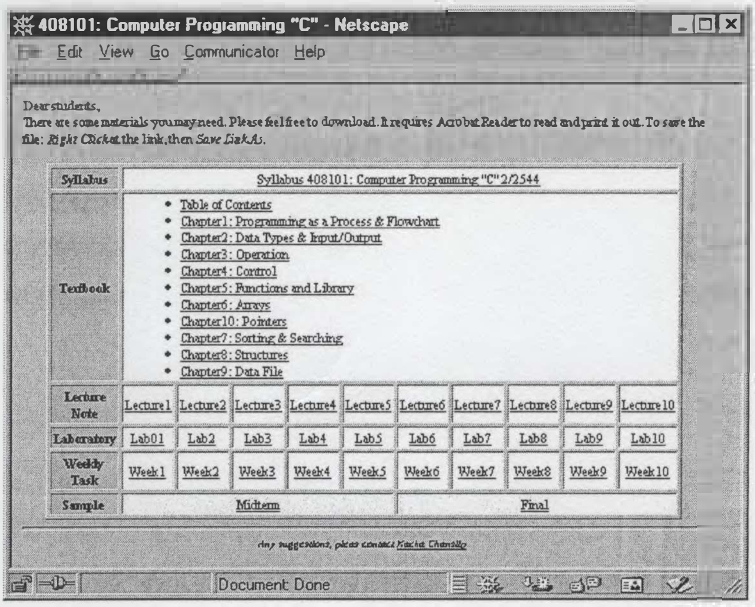


Figure 4.2: Syllabus & lecture page

The DIVTIC program was designed so that when a student clicked on the Syllabus & Lecture Notes section, a browser would open a new window containing six different parts: Syllabus, Textbook, Lecture Notes, Laboratory, Weekly Task, and Sample (see Figure 4.2). Each section was either in the form of a PDF file or a compressed ZIP file that could be viewed via the browser or downloaded onto the computer. A description of each section is given as follows:

The ***Syllabus*** section includes information on the semester topic, the laboratory test date, the midterm and the final test dates, and relevant course textbooks. This is available as a PDF.

The ***Textbook*** section includes a revised version of the notes and papers produced by the researcher in the previous 3 years of this study. It was divided into 10 chapters to make it easy for students to download or view in a browser. It is available as a PDF.

The ***Lecture Notes*** section contains Powerpoint slides used by the instructor in the lectures. Under normal conditions, these were to be uploaded onto the Web once a week prior to the commencement of the lecture. Thus, the subjects could download and view the lecture notes ahead of time. These resources were made available in DIVTIC as a compressed ZIP file which facilitated uploading onto and downloading from the server.

The *Laboratory* section was designed to provide all students who were registered in the course with a download of the laboratory weekly problem ahead of time. It was also a place for students who did not have the laboratory sessions and who did not have the laboratory weekly problem. This feature was designed to ensure that all the students had the laboratory weekly problem on hand before coming to the laboratory session, hopefully encouraging them to solve the problem in advance by writing some source code either on a piece of paper or in a text editor program.

The *Weekly Task* section was designed to contain weekly activities that the students in the experimental group would be required to do at the beginning of each laboratory session. Each activity was comprised of two "fill-in" questions which would take about 30-45 minutes to complete.

The *Sample* section was designed to contain a sample of previous midterm and final examinations. The midterm examination contained 25 multiple-choice questions and the final examination contained 50 multiple-choice questions.

4.1.2 Computer Structure

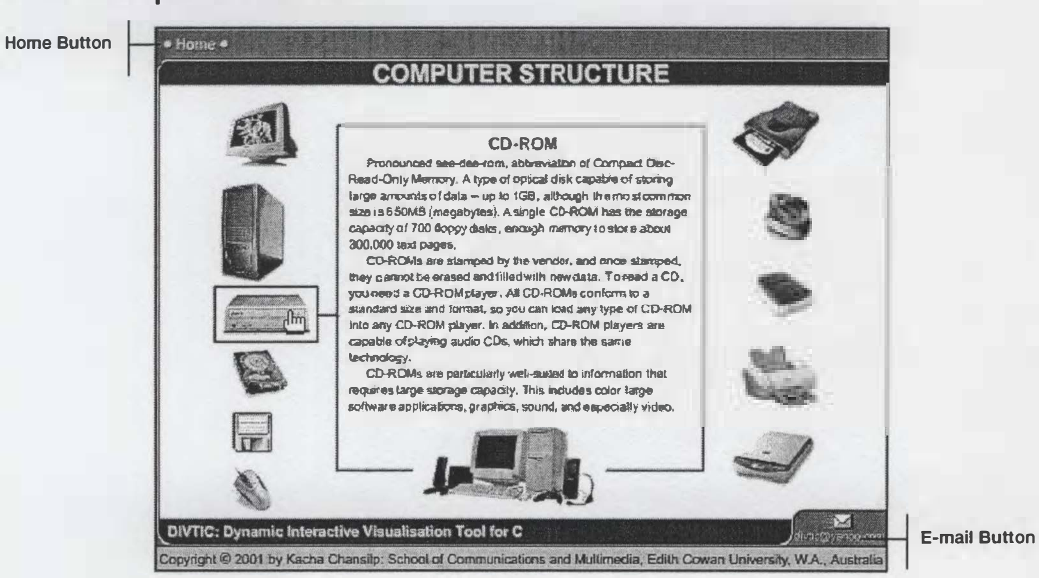


Figure 4.3: Computer structure page

The Computer Structure page was designed to offer an overview of the basic structure of a computer that contained 12 different parts of hardware including a monitor, central processing unit [CPU], CD-ROM, hard disk, floppy disk, mouse, computer system, scanner, printer, modem, digital camera and zip drive. Each part was given a definition

and description that appeared when a student moved the mouse over each hardware icon. The related information was displayed in the middle of a rectangle box. This section was intended to be useful for the students who were new to computers or had no idea about computers and their parts (see Figure 4.3). A Home button, at the top left-hand corner, was provided for the students to link back to DIVTIC's home page. An E-mail button was also provided at the bottom right-hand corner so that the students could easily send an e-mail to the researcher by clicking on it to activate an e-mail program.

4.1.3 Animated Examples

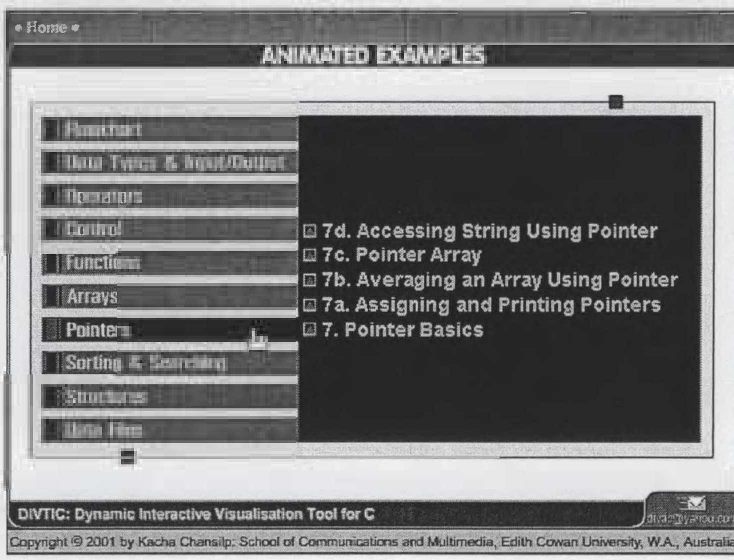


Figure 4.4: Animated examples page

The Animated Examples page was designed to contain 10 chapters. Each chapter comprised three to six animated examples. The students needed to move the mouse over each chapter heading to reveal a submenu linking to some animated examples. For example, in Figure 4.4, a student using DIVTIC would move the mouse over a chapter called Pointers on the left-hand side and the submenu would display on the right-hand side containing links to five different animated examples. The student could then move the mouse over a specific animated example and click on it to activate it.

As the Animated Examples section was the most important section, there is a need to describe its features in more detail. For example, when students click on the "7c. Pointer Array", a new window displays (see Figure 4.5) and so on. The top left-hand side is a panel for a given problem to be displayed with an icon at the bottom called C Source Code. When students click on this icon, the given problem changes to the C source code

(see Figure 4.6). This feature was incorporated because it could allow the user to pause and think about the given problem and to try to write the code before clicking the icon to see the result. This was planned to encourage the students to be more active learners.

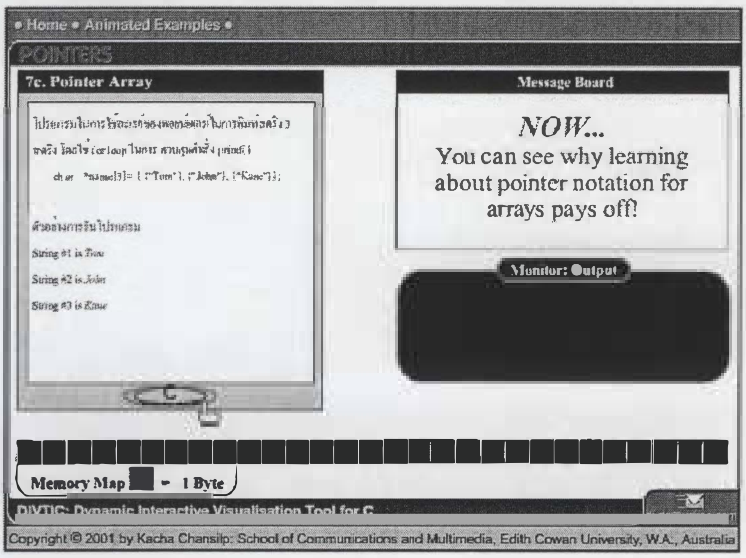


Figure 4.5: A sample of animated example 1

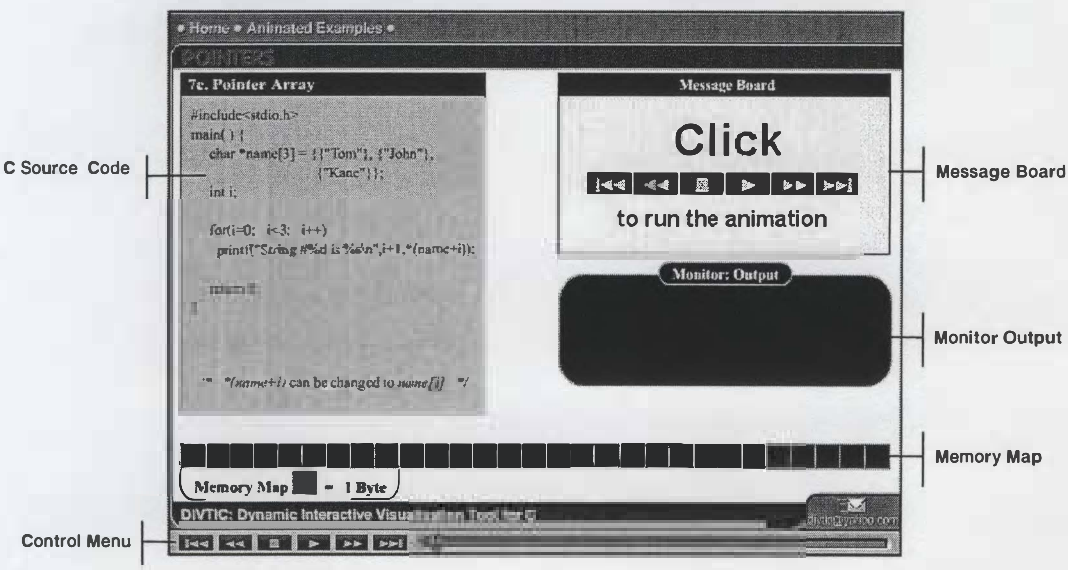


Figure 4.6: A sample of animated example 2

The top right-hand side has another panel called the Message Board. Its purpose is to display dynamic messages at any specific time when the marker in the Source Code panel runs past any particular commands (see Figure 4.7).

When the students compile and run their source code via the C compiler, they see the output after the whole program has been executed. The output command in the source

code can be more than one command depending on how many outputs a user needs. However, as it is not synchronous it may lead users to misunderstand the concept. In order to ensure that this does not happen, the panel below the Message Board, called the Monitor Output, synchronously demonstrates the real-time output (see Figure 4.7)

Another panel located under the Source Code and Monitor Output panels is the Memory Map (see Figure 4.7). This was designed to help the users develop mental models of how a computer stores data and its values. When the marker runs through the declaration section, the equivalent section in the memory map is highlighted. The number of boxes is dependent on the type of that particular variable, and is individually assigned to that variable.

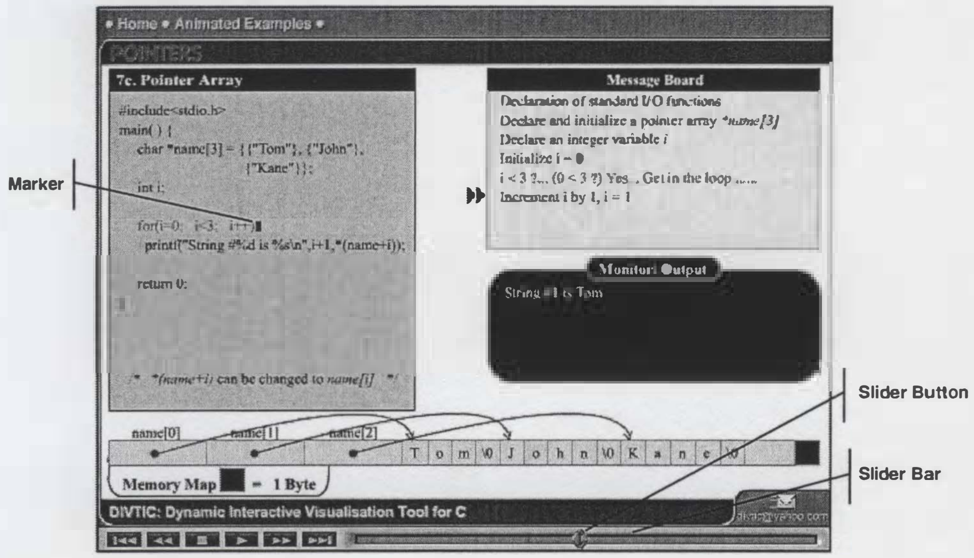


Figure 4.7: A sample of animated example 3

The last handy tool is the Control Menu. It is at the bottom of the screen (see Figure 4.7) and appears when the source code is displayed. It was designed to control the animation. It works the same way as the video controller buttons. It comprises six static buttons, one slider bar and one dynamic slider button that can be dragged into any position on the slider bar to see any particular spot of the animation. It is shown in Figure 4.8.

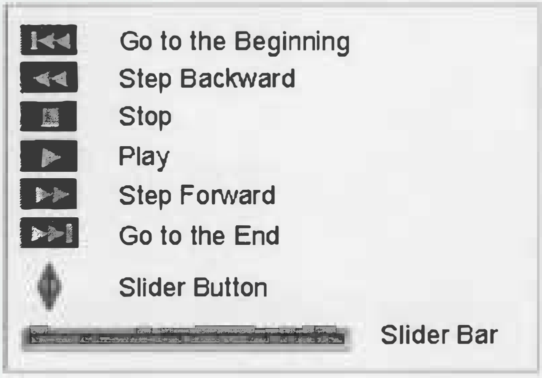


Figure 4.8: Control menu

4.1.4 C Compiler

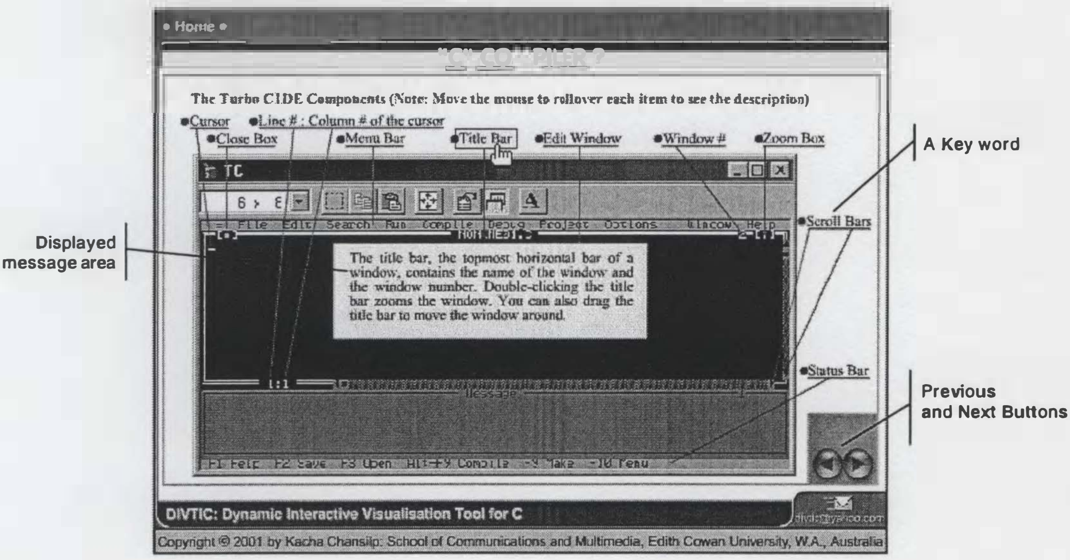


Figure 4.9: C compiler page 4

The C compiler page was designed to teach users how to use C compiler step-by-step in eight pages. Figure 4.9 shows the interaction when user has moved the mouse over any particular key word: the associated description is displayed in the middle of the screen. In this case, it is a Title Bar. This feature also includes two buttons, previous and next, to go back to the previous page and move on the next page.

4.1.5 C WebBoard

This was designed to permit students to have asynchronous discussion so that they could share ideas, information, questions and suggestions by reading and posting the messages at whatever hour of the day they were most productive. The instructor would also visit the WebBoard page on a daily basis to reply to the messages. This would

ensure that all messages were answered by either a peer or the instructor himself (see Figure 4.10).

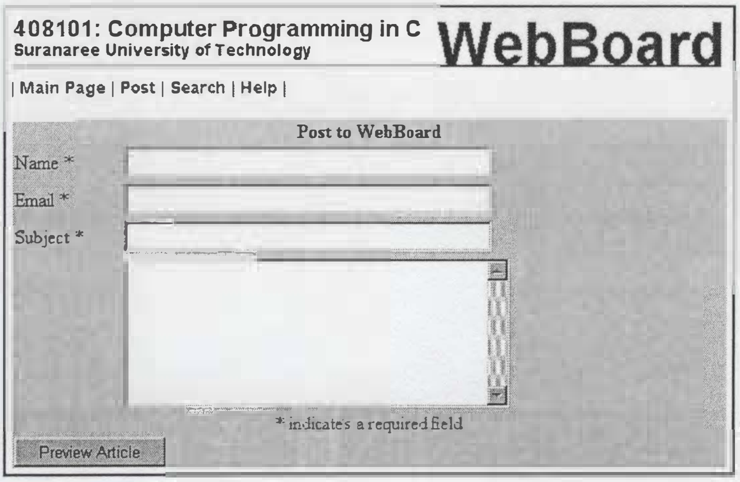


Figure 4.10: WebBoard page

4.1.6 Self-Evaluation

Figure 4.11 shows the Self-Evaluation page which also covers 10 chapters. When the students click, for example, on Chapter 2, Data Type & Input/Output, a new HTML page opens on a browser showing several multiple-choice questions. This can be seen in Figure 4.12.

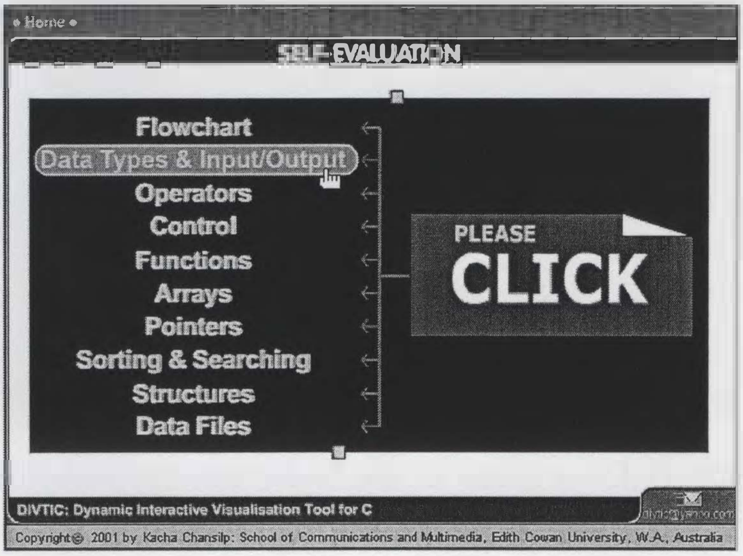


Figure 4.11: Self-evaluation page

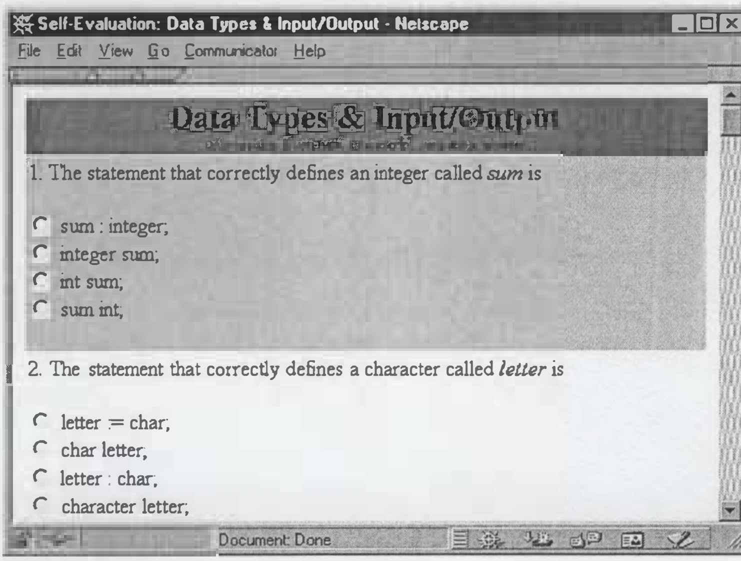


Figure 4.12: Data type & input/output sample 1

The user can then do the self-assessment test by themselves by selecting any answer. JavaScript was used to create an immediate feedback feature that pops-up right after an answer has been selected as shown in Figure 4.13.

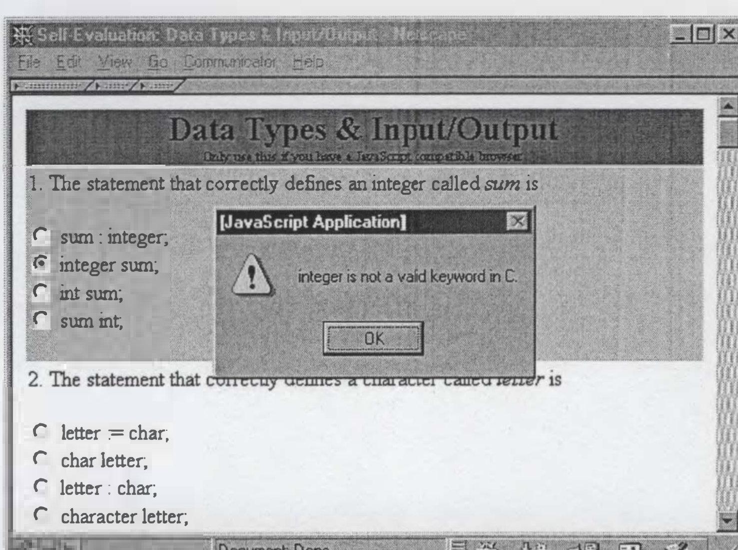


Figure 4.13: Data type & input/output sample 2

4.1.7 FAQ Pool

The FAQ Pool page was designed to group frequently asked questions into related chapters. When students click on any chapter in the left-hand side panel, the appropriate frequently asked questions are then displayed on the right-hand side next to the chapter panel with two scroll down and scroll up buttons (see Figure 4.14).

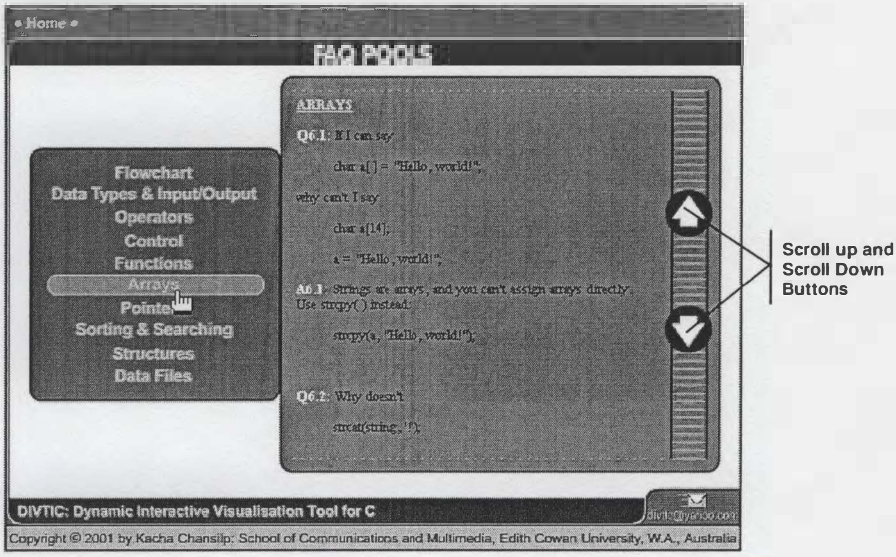


Figure 4.14: FAQ pool page

4.1.8 C References & Links

There has been a substantial amount of information written and published about the C programming language. This information can provide students with interesting insights into the language and its variances. The “C References & Links” section was designed to provide students with access to this information (Figure 4.15).

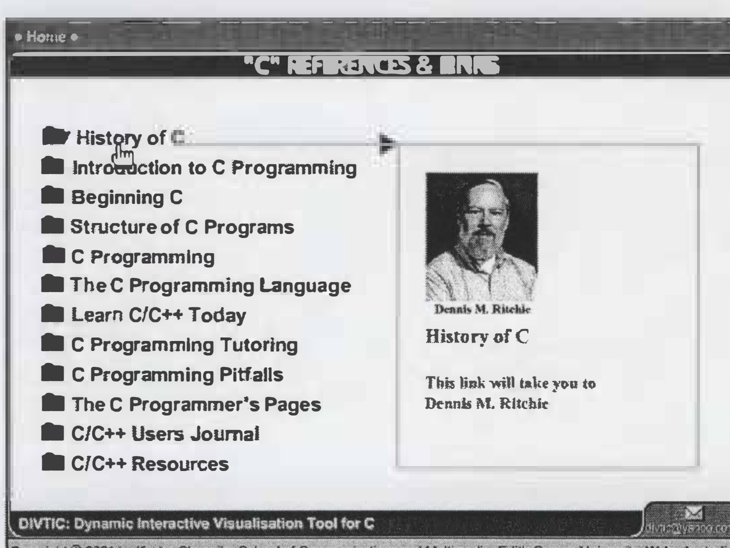


Figure 4.15: C references & links page

4.2 DIVTIC Set up Website

There were two different websites developed for this study. One was created by the instructor providing some course relevant information. It was designed for all students

to access. The other website was DIVTIC's website comprising the same information plus some extra materials from the DIVTIC system. This website was designed only for the experimental group to access.

The set up of the server for DIVTIC's website was divided into four categories: hardware specification, software specification, log in strategy, and file structure.

4.2.1 Hardware Specification

The hardware for the server had the following specifications:

- CPU: Pentium MMX 233 MHz,
- SDRAM: 96 MB,
- Hard Disk: 4 GB,
- NIC (Network Interfacing Card): PCI 82557 (Ethernet Pro 100) by Intel Corporation,
- IP: 202.21.140.172
- URL: http://202.21.140.172/divtic.html

4.2.2 Software Specification

The software requirements to set up the server were as follows:

- **OS:** Linux Mandrake 8.0 (Traktopel) – Kernel Version 2.4.3,
- **Web Server:** Apache-Advanced Extranet Server Version 1.3.19 (Linux-Mandrake/3mdk) Built: April 16, 2001 07:50:02,
- **Perl Script:** This was used to record users' log-in time,
- **E-mail Program:** When the student clicks on the icon "Send mail to [redacted] it opens a file, "email.html", and calls a file, "smkacha.cgi" which resides in /home/kacha/public_html/cgi-bin. File, "smkacha.cgi", then sends the mail to [redacted] and
- **WebBoard Program:** When the student clicks on the C WebBoard icon in the main page of DIVTIC program, it calls a file "wbkacha.cgi", which resides in /home/kacha/public_html/cgi-bin/. File "wbkacha.cgi", then opens a file called "wbkachavar.pl" which contains configurations for the WebBoard

program. File "wbkacha.cgi" also connects to a database called "dbkacha", which resides in the Mysql DataBase Server. All WebBoard data is kept in the dbkacha database for convenient access.

4.2.3 Log in Strategy

The system required a facility to track student usage so a log in system was developed to facilitate this. The log in procedure is described below:

1. The user logs into the system, file "divtic.html" through a Web URL residing in /var/www/html/ which is then refreshed to http://202.21.140.172/~kacha/index.html.
2. File "index.html" in /home/kacha/public_html/index.html asks for the User Name and Password, then it calls a cgi script, "pwd.pl", which resides in /htmo/kacha/public_html/cgi-bin/.
3. The cgi script, "pwd.pl", receives the data, User Name and Password, from index.html and checks it with a file "pword_divtic.lst" to validate the User Name and Password.
4. If it is a valid user, it then refreshes to /home/kacha/public_html/Goodluck.html and also opens the first page of the Divtic Program, "start_run_divtic.html", which is in /home/kacha/public_html/bxxxxxxx/, and where bxxxxxxx is the user name. If it is invalid, then it prompts an invalid User Name or Password and allows the user to try again (see Figure 4.16).

The log in process is recorded by pwd.pl in a file called "pwdlog_divtic.txt", which resides in /home/kacha/public_html/cgi-bin/, regardless of whether or not it is successful. This feature was designed to help keep track of the time spent in DIVTIC. It was developed to record the time spent in each module by each student writing relevant information to log files for later retrieval purposes.

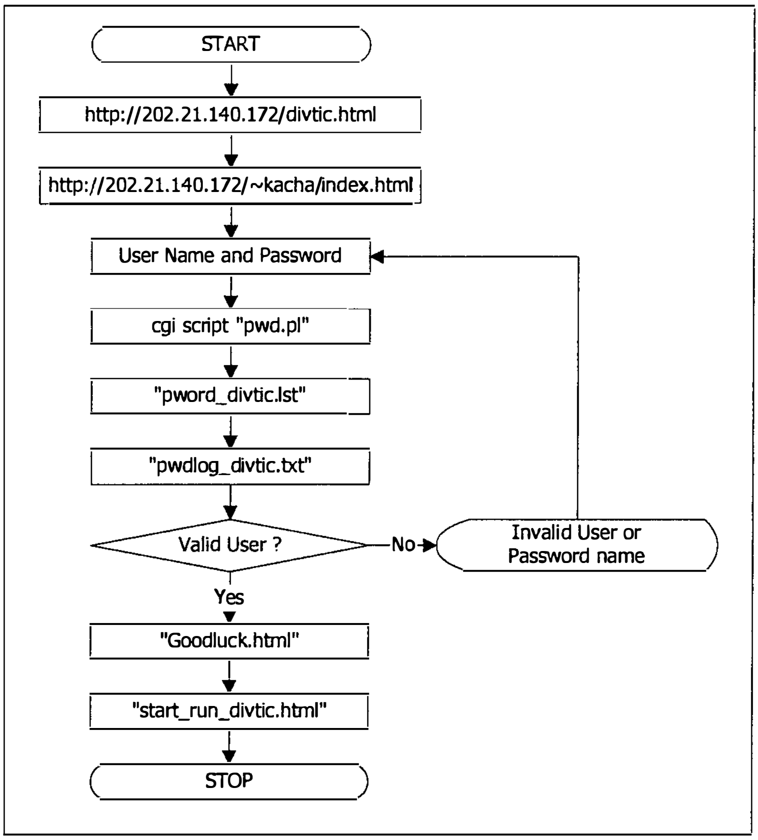


Figure 4.16: Log-in flowchart

4.2.4 File Structure

All related DIVTIC files were located in /home/kacha/public_html/All. The total size of all the files was 9.8 MB. The system was designed so that each user had their own folder by using their user name as the name of their folder. For example, if the user name is "b4111111", then a folder called "b4111111" is created to reside in /home/kacha/public_html/. In each user folder, there are only the soft links that point to the original DIVTIC files in /home/kacha/public_html/All. Each user's folder is only 4KB. The reason for this is as follows:

1. To reduce the size in each user's folder from 9.8 MB to 4KB;
2. To enable simple modification of the original DIVTIC files if necessary; and,
3. To keep a record of each user to see when each user had logged in, the files used, and length of use.

To create each user folder name in /home/kacha/public_html/, a bash shell script called "chang_create_dir" is used. To run this bash shell script, a period and a slash is required in front of the file (./chang_create_dir). The bash shell script opens and reads a file

called "pword_divtic.lst" in "/home/kacha/public_html/cgi-bin/" and creates each associated folder automatically in /home/kacha/public_html/. It then reads all file names in /home/kacha/public_html/All and creates soft links in each user folder to be pointed to /home/kacha/public_html/All (see Figure 4.17).

A bash script, "chang_backup", keeps a record of two things:

1. File Ldate.tgz keeps a log of log in user produced by pwd.pl, an access log of web server, and an error log of web server.
2. File Wdate.tgz keeps all web pages in /home/kacha/public_html/.

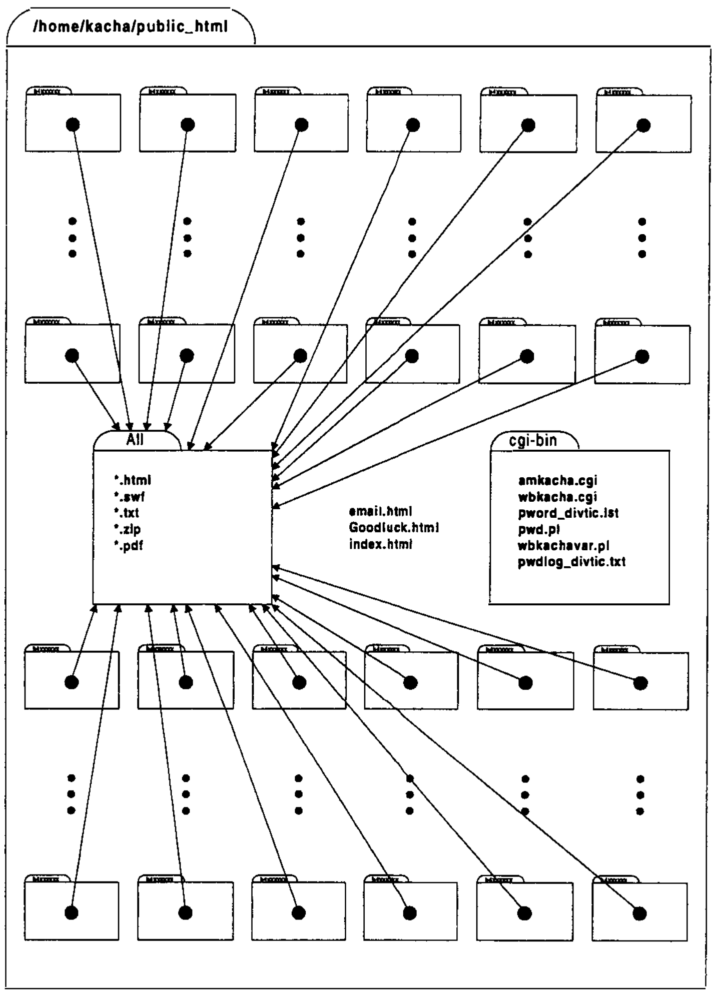


Figure 4.17: DIVTIC's website structure

With the system planned and developed, a pilot study was carried out ahead of the full study to enable the system to be trialed and any problems identified and fixed ahead of the full implementation.

4.3 The Pilot Study

The Pilot study was an important step in the development process of DIVTIC. It was used to test and focus on specific areas of the usability and technical robustness of DIVTIC ahead of the main study. It was conducted at Suranaree University of Technology (SUT) in Thailand where the final experiment would be conducted. It was undertaken at the beginning of August 2001.

The pilot study was specifically restricted to the use of the Animated Examples section which was the major and most important section for this research. Its purpose was to explore the use of the Animated Examples section by checking its usability, data accuracy, and technical elements.

There were 46 animated examples altogether. Twelve subjects were involved in the pilot study to run and test the animations. The Pilot Study—Self-administered Evaluation Questionnaire form (Figure 4.18 in section 4.3.2.1) was given to subjects. They were asked to fill in the form after testing each animated example. The form contained six short questions using a five-point Likert rating scale and two open-ended questions. The data from this form was used to evaluate and implement the feasibility and usability of DIVTIC. Modifications to DIVTIC were made in terms of grammar and concept correction and are described later in the following section.

4.3.1 Pilot Study Setting

Advertisements were posted on the bulletin board seeking six undergraduate engineering students who had never taken the course 408101 Computer Programming. The subjects were not the same as those in the main study, however they were required to have the same level of experience and knowledge. The pilot study also included six experienced tutors in the C programming language who were the tutors throughout the experimental study period. The students and tutors were paid a small honorarium as an encouragement and a reward for their participation.

Due to the large number of animations to be checked, subjects in the pilot study were also given a selection of animations to check. The 46 animated examples were divided equally in terms of difficulty into three parts: Part A, Part B, and Part C (see Table 4.1). Each part contained only one executable file including all associated files, which fitted

onto one floppy disk. In this way, each part was copied and duplicated onto four floppy disks in total.

The total number of the subjects in the pilot study was 12. These were divided into three groups: Group1, Group2, and Group3. Each group was comprised of two students and two tutors. Every subject was provided with a floppy disk containing either Part A, Part B or Part C, and the Pilot Study—Self-administered Evaluation Questionnaire form. The Pilot Study—Self-administered Evaluation Questionnaire form was comprised of two parts: Part 1 and Part 2. Part 1 was a questionnaire containing eight short questions using a five-point Likert rating scale. Part 2 was an open-ended question which contained two parts. The subjects were asked to test and check each animated example carefully and complete the Pilot Study—Self-administered Evaluation Questionnaire form after each individual test. Each subject took approximately 4 hours to complete checking all the given animated examples and filling in the Pilot Study—Self-administered Evaluation Questionnaire forms.

Table 4.1: Grouping animated example files for the pilot study

Part A: Group1	**Part B: Group2**	**Part C: Group3**
1. Flowchart Symbols	4c. Hollow Triangle	8. Sorting & Searching Overview
1a. Credit Approval	5. Functions and Library	8a. Numeric Sorting
1b. Circle in the Square	5a. Lucky Number	8b. String Sorting
1c. Menu Display	5b. Double Character	8c. Linear Search
1d. Samples	5c. Reverse String	8d. Binary Search
2. Data Basics	6. Arrays?	9. Structure Basics
2a. Input/Output *using scanf( ) and printf( )*	6a. Array Summation	9a. Creating a Basic Structure
2b. Input/Output *using getchar( ), putchar( ), gets( ), and puts( )*	6b. Vowel Counter	9b. Pointer and an Array of Structure
2c. Area of Triangle	7. Pointer Basics	9c. Pointer, Function, and Structure
2d. Triangular Circumferences	7a. Assigning and Printing Pointers	10. Data File?
3. Operators	7b. Averaging an Array Using Pointer	10a. Writing a Simple Text File
3a. Basic Arithmetic Operator	7c. Pointer Array	10b. Reading a Text File
3b. Condition Operator	7d. Accessing String Using Pointer	10c. Appending a Text File
3c. Increment/Decrement Operator		10d. Writing a Simple Binary File
3d. Dividing a Number		10e. Reading a Binary File
4. General control Forms		
4a. Print value of I from 1 to 3		
4b. Averaging numbers		

The tutors were given one week to complete the study. They did the test in their own time and returned both the floppy disk and the Pilot Study—Self-administered Evaluation Questionnaire form to the researcher.

In addition, each student was asked to be available to test the animation in a different time slot in the researcher's office. There were six time slots and each time slot lasted 4 hours. This process was designed to ensure that each student had done all the animation tests on their own. All the feedback from the Pilot Study—Self-administered Evaluation

Questionnaire form was used to enhance the usability and feasibility of DIVTIC which is described in the next section.

4.3.2 Pilot Study Findings

The pilot study was completed by the middle of August 2001. A number of changes were made to DIVTIC on the basis of the feedback as described in the following section.

4.3.2.1 Findings: Part 1

Part 1 in the Pilot Study—Self-administered Evaluation Questionnaire form used a Likert scale rating, ranging from 1 to 5, which corresponded to strongly disagree, disagree, not applicable, agree, and strongly agree, respectively. It was designed to evaluate a subject' s impressions of the navigation, speed of access, level of reliability, and utility of the various features. It was comprised of eight short questions as shown in Figure 4.18 and given to the subjects to apply to each animation.

Pilot Study: Self-administered Evaluation Questionnaire form

Animation Title: ______________________________

Part 1: Please indicate how much you disagree or agree with each of the following statements.

Note: 1 = Strongly Disagree
2 = Disagree
3 = Not Applicable
4 = Agree
5 = Strongly Agree

Question	Strongly Disagree 1	Disagree 2	Not Applicable 3	Agree 4	Strongly Agree 5
The speed of the animation is okay.					
The animation is easy to understand in concept.					
The animation process is easy to control.					
The interface of the animation is good.					
The animation is easy to navigate.					
The message board is well explained in details.					
The memory map is well explained.					
The problem is too difficult.					

Figure 4.18: The pilot study—Self-administered evaluation questionnaire form—Part 1

The statistical software program, SPSS (Statistical Package for the Social Sciences), was used to calculate the mean from the feedback given by the subjects in order to explore the usability and technical accuracy of each animated example. In determining

the subjects' levels of satisfaction of the various elements, it was interpreted that a mid range score between 2.75 to 3.25 was indicative of a non-applicable result; a score less than 2.75 was indicative of an unsatisfactory result; and a score greater than 3.25 was indicative of a satisfactory result. To determine the difficulty of the problem, the mid range score between 2.75 to 3.25 was indicative of a non-applicable result; a score less than 2.75 was an indication of an easy problem; and a score greater than 3.25 was an indication of a difficult problem. The memory map section was not incorporated into all the animations including all animations in Chapter 1—Flowchart as well as animations 2, 3, 4, 5, 6, 7, 8, 9, and 10 as they contained fundamental information. The following discussion of the findings is organised on a chapter-by-chapter basis.

Table 4.2: The mean of each animation and average of mean of each question in Chapter 1—Flowchart

Animation	Mean							
	Speed	Concept	Control	Interface	Navigation	Message Board	Memory Map	Difficulty
1	4.50	4.75	3.75	3.50	3.25	4.00	-	2.50
1a	3.50	3.75	4.00	4.00	3.00	4.00	-	1.50
1b	3.25	4.25	4.25	3.75	3.25	4.00	-	1.75
1c	3.50	3.50	4.00	3.75	3.50	3.75	-	3.00
1d	4.25	3.50	4.25	3.50	4.00	2.75	-	3.25
Average Mean	***3.80***	***3.95***	***4.05***	***3.70***	***3.40***	***3.70***	-	***2.35***

Table 4.2 shows the mean of each question for each animated example in chapter 1—Flowchart. The results revealed that the subjects were generally satisfied by the speed of the animation (average mean = 3.80), ease of understanding of the concept (average mean = 3.95), and the ease of controlling the animation process (average mean = 4.05). The interface of the animation and message board explanation were also to the subjects' satisfaction (average mean = 3.70). The ease of the navigation also achieved a positive response (average mean = 3.40). However, the average mean of the problem difficulty (2.35) revealed that subjects disagreed in the statement that the problem was too difficult. All problems were deemed to be easy except for problems lc (m = 3.00) and 2d (m = 3.25) where subjects responded with non-applicable results. The speed of the animation 1b had a mean of 3.25 which was also indicative of a non-applicable result. The ease of understanding of the concept of animation 1 had a mean of 4.75 which was indicative of a highly satisfactory result, while the mean of the interface of the animation (average mean = 3.70) and the ease of navigation (average mean = 3.40) appeared to produce consistently satisfactory results. The explanation in the message

board of animation 1d had a mean of 2.75 which appeared to be a non-applicable result. There was no comment on the memory map since it was not included in Chapter 1.

Table 4.3: The mean of each animation and average of mean of each question in Chapter 2—Data Types & Input/Output

Animation	Mean							
	Speed	Concept	Control	Interface	Navigation	Message Board	Memory Map	Difficulty
2	4.50	4.00	3.75	3.50	3.25	3.25	-	2.00
2a	3.50	4.50	4.25	3.75	3.75	4.00	3.75	2.75
2b	3.75	4.50	4.25	3.75	3.75	4.00	3.75	2.50
2c	3.75	5.00	4.25	3.75	3.75	3.75	3.75	1.75
2d	3.75	4.25	4.25	3.75	3.75	3.75	3.75	3.50
Average Mean	***3.85***	***4.45***	***4.15***	***3.70***	***3.65***	***3.75***	***3.75***	***2.56***

Table 4.3 shows an increase in positive responses. The speed of the animation (average mean = 3.85) resulted in a consistently satisfactory result except for animation 2 (m = 4.50), which resulted in a highly satisfactory result. The ease of understanding of the concept (average mean = 4.45) was indicative of a highly satisfactory result, especially in animation 2c (m = 5.00), which appeared to give the highest result. The ease of controlling the animation process (average mean = 4.15), the interface of the animation (average mean = 3.70), and the ease of navigation (average mean = 3.65) also appeared to produce consistently satisfactory results, except for the ease of navigation of animation 2 (m = 3.25) which gave a non-applicable result. The explanation in the message board (average mean = 3.75) indicated a satisfactory result except for animation 2 (m = 3.25), which was a non-applicable result. The explanation in the memory map of each animation had a consistent mean (m = 3.75), which was indicative of a satisfactory result. The difficulty of the problem (average mean = 2.56) appeared to be easy especially for problem 2c (m = 1.75), which appeared to be the easiest one. However, the responses for problem 2d (m = 3.50) demonstrated that it was difficult. Overall, the results revealed that the subjects were generally satisfied with all aspects of the animation.

Table 4.4: The mean of each animation and average of mean of each question in Chapter 3—Operators

Animation	Mean							
	Speed	Concept	Control	Interface	Navigation	Message Board	Memory Map	Difficulty
3	4.50	4.25	4.00	4.00	3.75	4.00	-	2.50
3a	4.25	4.50	4.00	4.25	3.75	4.25	4.00	1.50
3b	4.00	4.50	4.25	4.25	4.00	4.25	4.00	1.00
3c	3.50	4.50	4.25	4.25	4.00	4.00	4.00	1.75
3d	3.50	3.75	4.00	4.00	4.00	3.25	4.00	2.50
Average Mean	*3.95*	*4.30*	*4.10*	*4.15*	*3.90*	*3.95*	*4.00*	*1.89*

Table 4.4 shows positive responses. The speed of the animation (average mean = 3.95) appeared to be satisfactory. The ease of understanding of the concept (average mean = 4.30) was highly satisfactory. However, the mean of animation 3d (m = 3.75) was the lowest mean and appeared to be difficult to understand. This was to be expected, since the concepts were increasing in complexity. The ease of controlling the animation process (average mean = 4.10), the interface of the animation (average mean = 4.15), and the ease of navigation (average mean = 3.90) gave consistently satisfactory results. The explanation in the message board (average mean = 3.95) was satisfactory except for animation 3d (m = 3.25) which indicated a non-applicable result. The explanation in the memory map of each animation had a consistent mean (m = 4.00) which was indicative of a satisfactory result. The difficulty of the problem (average mean = 1.89) appeared to be easy, especially problem 3b (m = 1.00). Overall, the results revealed that the subjects were generally satisfied with all aspects of the animation.

Table 4.5: The mean of each animation and average of mean of each question in Chapter 4—Control

Animation	Mean							
	Speed	Concept	Control	Interface	Navigation	Message Board	Memory Map	Difficulty
4	4.50	4.25	4.00	4.25	3.50	3.75	-	2.00
4a	3.50	4.25	4.00	4.25	4.00	4.25	4.00	2.00
4b	3.25	4.00	4.00	4.25	4.00	4.25	4.25	2.75
4c	3.33	4.00	3.67	4.00	3.33	4.33	4.33	3.33
Average Mean	*3.67*	*4.13*	*3.93*	*4.20*	*3.73*	*4.13*	*4.20*	*2.54*

Table 4.5 shows positive responses. The speed of the animation (average mean = 3.67) produced a satisfactory result except for animation 4b (m = 3.25), which was a non-applicable result. The ease of understanding of the concept (average mean = 4.13) was indicative of a consistently satisfactory result. The ease of controlling the animation

process (average mean = 3.93) and the interface of the animation (average mean = 4.20) produced consistently satisfactory results. The average of the mean of the interface of the animation had increased from Chapter 1 (average mean = 3.70) through Chapter 2 (average mean = 3.70), to Chapter 3 (average mean = 4.15). This suggested that the subjects might have developed a greater appreciation of DIVTIC. The ease of navigation (average mean = 3.73) was satisfactory. The explanation in the message board (average mean = 4.13) was also indicative of a satisfactory result. The mean of each animation slightly increased from animation 4 (m = 3.75), through animations 4a and 4b (m = 4.25), to animation 4c (m = 4.33). This revealed that subjects were satisfied and understood more as they used DIVTIC more. The explanation in the memory map of each animation (average mean = 4.20) provided a highly satisfactory result. The average of the mean of the explanation in the memory map had slightly increased from Chapter 2 (average mean = 3.75) to Chapter 3 (average mean = 4.00). This also suggested that the subjects might have developed a greater appreciation of DIVTIC. The overall difficulty rating of the problem (average mean = 2.54) appeared to be easy except for problem 4c (m = 3.33), which was considered difficult as was to be expected, since the concepts were increasing in complexity. Overall, the results revealed that the subjects were generally satisfied with all aspects of the animation.

Table 4.6: The mean of each animation and average of mean of each question in Chapter 5—Functions

Animation	Mean							
	Speed	Concept	Control	Interface	Navigation	Message Board	Memory Map	Difficulty
5	3.75	3.75	4.25	4.00	4.25	4.00	-	2.50
5a	4.25	4.25	4.25	4.50	4.00	4.00	4.25	2.50
5b	4.25	4.00	4.00	4.00	4.25	4.25	4.00	3.00
5c	4.50	3.50	4.25	4.00	4.25	3.75	4.25	2.75
Average Mean	***4.19***	***3.87***	***4.19***	***4.13***	***4.19***	***4.00***	***4.17***	***2.69***

Table 4.6 shows further positive responses. The speed of the animation (average mean = 4.19) and the ease of understanding of the concept (average mean = 3.87) resulted in satisfactory results. However, the average mean of the ease of understanding of the concept (average mean = 3.87) had decreased from Chapter 2 (average mean = 4.45) through Chapter 3 (average mean = 4.30), to Chapter 4 (average mean = 4.13). This suggested that the concepts were getting more complicated and difficult to understand as more materials were incorporated, producing more tasks and information for the subjects to accommodate. The ease of controlling the animation process (average mean

= 4.19), the interface of the animation (average mean = 4.13), and the ease of navigation (average mean = 4.19) resulted in consistently satisfactory results. The explanation in the message board (average mean = 4.00) produced a satisfactory result. The explanation in the memory map (average mean = 4.17) produced a highly satisfactory result. The overall difficulty of the problem (average mean = 2.69) appeared to be easy. However, problems 5 and 5a had a mean of 2.50, which appeared to be easy problems while problems 5b (m = 3.00) and 5c (m = 2.75) appeared to be non-applicable. Therefore, the subjects still accepted the difficulty of the problem within acceptable boundaries. Overall, the results revealed that the subjects were generally satisfied with all aspects of the animation.

Table 4.7: The mean of each animation and average of mean of each question in Chapter 6—Arrays

Animation	Mean							
	Speed	Concept	Control	Interface	Navigation	Message Board	Memory Map	Difficulty
6	4.00	3.75	4.25	4.00	3.75	4.00	-	2.75
6a	4.00	3.75	4.50	4.00	4.00	4.25	4.00	2.50
6b	4.00	3.75	4.25	4.00	4.00	4.50	4.25	3.00
Average Mean	*4.00*	*3.75*	*4.33*	*4.00*	*3.92*	*4.25*	*4.12*	*2.75*

Table 4.7 again shows positive responses. The speed of the animation (average mean = 4.00), the ease of understanding of the concept (average mean = 3.75), the ease of controlling the animation process (average mean = 4.33), the interface of the animation (average mean = 4.00), and the ease of navigation (average mean = 3.92) produced consistently satisfactory results. The explanation in the message board (average mean = 4.25) provided a satisfactory result with the mean of each subsequent animation rapidly increasing from animation 6 (m = 4.00) through animation 6a (m = 4.25), to animation 6b (m = 4.50). This suggests that the subjects appeared to have more understanding as they became more accustomed to DIVTIC. The explanation in the memory map (average mean = 4.12) indicated a satisfactory result. The difficulty of the problem (average mean = 2.75) appeared to be a non-applicable result except for problem 6b (m = 2.50), which appeared to be easy, but the subjects still accepted the difficulty of the problem within acceptable bounds. Overall, the results were very consistent across the three animations and they revealed that the subjects were generally satisfied with all aspects of the animation.

Table 4.8: The mean of each animation and average of mean of each question in Chapter 7—Pointers

Animation	Mean							
	Speed	Concept	Control	Interface	Navigation	Message Board	Memory Map	Difficulty
7	4.25	3.25	3.75	4.00	4.00	3.75	-	3.50
7a	4.00	3.50	4.00	4.50	3.75	4.25	4.25	2.75
7b	4.50	3.75	4.25	4.25	4.25	4.25	4.50	3.00
7c	4.25	3.75	4.25	4.50	4.00	4.50	4.00	2.50
7d	4.00	4.00	4.25	4.25	3.75	4.00	4.00	2.25
Average Mean	***4.21***	***3.65***	***4.10***	***4.30***	***3.95***	***4.15***	***4.19***	***2.80***

Table 4.8 also shows positive responses. The speed of the animation (average mean = 4.21) produced a consistently satisfactory result. The ease of understanding of the concept (average mean = 3.65) was indicative of a satisfactory result and the mean of each animation increased slightly from animation 7 (m = 3.25) through animation 7a (m = 3.50), animations 7b and 7c (m = 3.75), and once more to animation 7d (m = 4.00). This suggested that the subjects gained more understanding of the concept as they progressed through this chapter. However, overall the average mean for the ease of understanding of the concept (average mean = 3.65) had decreased from previous chapters, Chapter 2 (average mean = 4.45), Chapter 3 (average mean = 4.30), Chapter 4 (average mean = 4.13), Chapter 5 (average mean 3.87), and Chapter 6 (average mean = 3.75). This suggested that the concepts were getting more complicated and difficult to understand as more materials were incorporated, producing more tasks and information for the subjects to perceive. The ease of controlling the animation process (average mean = 4.10) and the interface of the animation (average mean = 4.30) gave consistently satisfactory results. The ease of navigation (average mean = 3.95) and the explanation in the message board (average mean = 4.15) also produced satisfactory results, while the explanation in the memory map (average mean = 4.19) was an indication of a highly satisfactory result. The overall difficulty of the problem (average mean = 2.80) appeared to be a non-applicable result. However, problems 7c (m = 2.50) and 7d (m = 2.25) appeared to be easy while problem 7 (m = 3.50) appeared to be difficult as was to be expected, since the concepts of this chapter, Chapter 7—Pointers, were increasing in complexity and more difficult to understand. Overall, the results revealed that the subjects were generally satisfied with all aspects of the animation.

Table 4.9: The mean of each animation and average of mean of each question in Chapter 8—Sorting & Searching

Animation	Mean							
	Speed	Concept	Control	Interface	Navigation	Message Board	Memory Map	Difficulty
8	4.33	4.50	4.33	4.33	3.25	4.50	-	2.50
8a	4.50	4.75	4.25	4.25	4.00	4.00	4.25	3.00
8b	4.25	4.75	4.75	4.25	4.25	3.50	4.25	2.75
8c	4.25	4.50	4.50	4.50	4.50	4.00	4.25	2.75
8d	4.00	4.25	4.50	4.50	4.25	4.25	4.00	3.00
Average Mean	***4.26***	***4.55***	***4.47***	***4.35***	***4.05***	***4.05***	***4.19***	***2.83***

Table 4.9 shows positive responses. The speed of the animation (average mean = 4.26) and the ease of understanding of the concept (average mean = 4.55) were indicative of highly satisfactory results. The ease of controlling the animation process (average mean = 4.47) and the interface of the animation (average mean = 4.35) produced consistently satisfactory results. The average mean of the ease of understanding of the concept (average mean = 4.55) and of the ease of controlling the animation process (average mean = 4.47) had increased statistically significantly from Chapter 7 (3.65 and 4.10 respectively). This suggested that the subjects had a greater understanding of the concepts and more control over the animation process at this stage. The ease of navigation (average mean = 4.05) produced satisfactory results except for animation 8 (m = 3.25), which had a non-applicable result. The explanation in the message board (average mean = 4.05) produced a satisfactory result. The explanation in the memory map (average mean = 4.19) produced a consistently satisfactory result. The difficulty of the problem (average mean = 2.83) generated a non-applicable result in all except for problem 8 (m = 2.50) which was judged as easy, while subjects still considered the difficulty of the problem within acceptable bounds. Overall, the results revealed that the subjects were generally satisfied with all aspects of the animation.

Table 4.10: The mean of each animation and average of mean of each question in Chapter 9—Structure

Animation	Mean							
	Speed	Concept	Control	Interface	Navigation	Message Board	Memory Map	Difficulty
9	4.33	4.33	4.67	4.50	4.00	4.25	-	2.50
9a	4.25	4.75	4.50	4.00	4.25	4.25	4.00	2.75
9b	4.25	4.50	4.75	4.50	4.25	3.75	4.00	3.25
9c	4.00	4.25	4.25	4.00	4.00	3.75	4.00	3.75
Average Mean	***4.20***	***4.47***	***4.53***	***4.25***	***4.13***	***4.00***	***4.00***	***3.14***

Table 4.10 shows positive responses. The speed of the animation (average mean = 4.20), the ease of understanding of the concept (average mean = 4.47), the ease of controlling the animation process (average mean = 4.53), and the interface of the animation (average mean = 4.25) were indicative of highly satisfactory results. The average mean of the ease of controlling the animation process (average mean = 4.53) had increased slightly from Chapter 8 (average mean = 4.47). This suggested that the subjects had more control over the animation process. The ease of navigation (average mean = 4.13) produced a consistently satisfactory result. The explanation in the message board (average mean = 4.00) produced a satisfactory result. The explanation in the memory map (average mean = 4.00) produced a consistently satisfactory result. The difficulty of the problem (average mean = 3.14) produced a non-applicable result. However, problem 9 (m = 2.50) appeared to be easy while problem 9c (m = 3.75) appeared to be a difficult problem. The mean for the difficulty of each animation had increased slightly from animation 9 (m = 2.50) through animation 9a (m = 2.75), to animation 9b (m = 3.25), and once more to animation 9c (m = 3.75). However, the subjects still accepted the difficulty of the problem as within acceptable bounds. Overall, it revealed that the subjects were generally satisfied with all aspects of the animation.

Table 4.11: The mean of each animation and average of mean of each question in Chapter 10—Data File

Animation	Mean							
	Speed	Concept	Control	Interface	Navigation	Message Board	Memory Map	Difficulty
10	4.33	4.25	4.33	3.75	3.75	4.50	-	4.00
10a	4.25	4.50	4.25	4.50	4.00	4.50	4.00	3.50
10b	4.25	4.50	4.25	4.25	4.25	4.25	4.50	3.25
10c	4.25	4.25	4.50	4.25	3.75	4.00	4.50	3.50
10d	4.25	4.00	4.25	4.25	4.25	4.25	4.00	3.75
10e	4.25	4.25	4.25	4.50	4.00	4.00	4.00	3.50
Average Mean	***4.26***	***4.29***	***4.30***	***4.25***	***4.00***	***4.25***	***4.20***	***3.55***

Table 4.11 shows further positive responses. The speed of the animation (average mean = 4.26), the ease of understanding of the concept (average mean = 4.29), and the ease of controlling the animation process (average mean = 4.30) produced consistently satisfactory results. The interface of the animation (average mean = 4.25) and the ease of navigation (average mean = 4.00) produced satisfactory results. The explanation in the message board (average mean = 4.25) and the explanation in the memory map

(average mean = 4.20) produced consistently satisfactory results. The difficulty of the problem (average mean = 3.55) appeared consistently difficult except for problem 10b (m = 3.25) which produced a non-applicable result. The average mean of the difficulty of the problem for each chapter had rapidly increased from previous chapters: the results for Chapters 1 to 10 were 2.35, 2.56, 1.89, 2.54, 2.69, 2.75, 2.80, 2.83, 3.14, and 3.55. This suggested that the level of difficulty of the problems grew more complicated and difficult as more materials were incorporated producing more tasks and information for the subjects to comprehend. This was to be expected since the concepts were increasing in complexity. Overall, the results revealed that the subjects were generally satisfied with all aspects of the animation except the difficulty of problem.

Table 4.12: The average of mean for all animated examples covered Chapters 1 to 10

Mean							
Speed	**Concept**	**Control**	**Interface**	**Navigation**	**Message Board**	**Memory Map**	**Difficulty**
4.04	4.16	4.21	4.10	3.85	4.02	4.05	2.73

In conclusion to Part 1, Table 4.12 shows that the average means of questions 1 to 7, from the speed of the animation to the explanation in the memory map, were typically high (3.85 to 4.16). This suggested that the subjects were satisfied with the functions of the animated examples. On the other hand, the average of the mean of question 8 (m = 2.73 indicated the difficulty of problem was rather low, but the difficulty of the problem for each chapter had increased rapidly from Chapter 1 through 10. This suggests that the difficulty of each problem increases as the materials become more complex. Figure 4.19 shows a line chart of the average of the mean for each question.

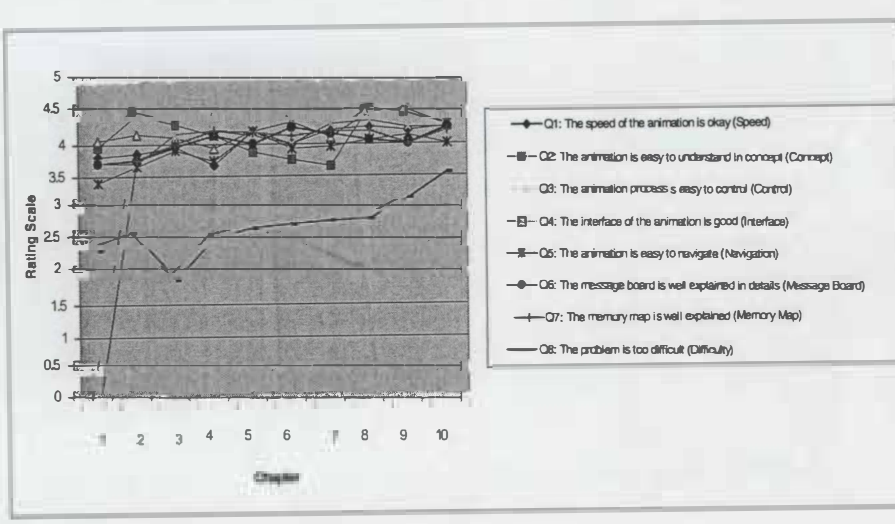


Figure 4.19: Line chart showing the average of the mean of each question

4.3.2.2 Findings: Part 2

Part 2 was comprised of two open-end questions as shown in Figure 4.20.

Part2: Please fill in the following questions

1. Is the animation accurate in all respects? Yes ❑ No ❑
   If No, please explain
   ____
   ____

2. Please suggest any improvements that will make this a better example
   ____
   ____

Figure 4.20: The pilot study—Self-administered evaluation questionnaire form—Part 2

The discussion of the findings from question 1 and 2 was categorised into six different parts: memory map, message board, interface/navigation, concept, spelling, and other. The feedback is presented in the following table (Table 4.13).

Table 4.13: Pilot study feedback from the subjects

Type	Details
Memory Map	• Memory map should have the same format by having the variable name on top of the memory location • Memory map is too small in chapter 9 and 10
Message Board	• Text is too small • Text should not blink • Should have a message when finished writing a flowchart • Should have a little pause when displaying the message to let user think along • Message should be in Thai
Interface & Navigation	• Using a right click instead of holding down the mouse to see the sample. • Flowchart needed to be more attractive • Text is too small, unsharp (blurred), need more contrast • Should have a button to direct what to do next once the animation finished playing. • The triangle bullet should be changed to another type of bullet • Should have a message telling where the control menu is located. • Inside each chapter, it should have an order number in front of each animated example, so the user knows which one comes first or after by using numbers 1, 2, 3, and so on or up and down arrows. • Memory map in the chapters 9 and 10 is too small • Play button in the control menu needed to be more accurate because the animation plays when user clicks anywhere on the page. • The position of the control menu is too low (at the bottom). • The animation seems to be too fast in the chapter 7 Pointer since it is a complicated chapter • In 8d. and 9b., the speed should slow down because there are many variables involved. • Text should be in different colour between the keyword (int, float, printf( );, etc.) and variable name (e.g., avg, n, pass, str, etc.)

Type	Details
Concept	• In 3. Operator, it should take the same format when user clicks either the example in the message board panel will show up or not. • in 4., Should include the arrow to direct the user when the loop goes back to check the condition • In 1.Flowchart, the terminate symbol had a wrong description • In 4., there is no ***if—then—else*** in C • In 2., Heading of 2. Data Basics should be Data Basics not Data Types & Input/Output • In 2., There is an overlap of the text in the Output Monitor (2a). The output 3.25 should be displayed as 3.250000 • In 4b., the monitor output should be changed from 6.66 to 6.666666 or in the source code should be changed form %f to %.2f. • Mismatch in 5. at ***isalpha( )*** and need to explain more about buffer/unbuffer • In 5c.,10a, and 10c., function ***gets(str);*** should be executed before the user enters an input. • In 6., ***array_new[subscript];*** should be changed to ***array_new[array_size];*** , ***printf ("number[%d] = %d\n", i, i=1);*** should be changed to ***printf("number[%d] = %d\n", i, number[i]);***, and ***array_name[subscript 1][subscript 2]...[subscript N]*** should be changed to ***array_name [size 1][size 2]...[size N]*** • In 7b., the problem ***finding an average of array of integer*** should be changed to ***finding an average of integer in an array*** • In 7d., ***cp = &msg[6]; *cp = 'W';*** may change to ****(cp+6) = 'w';*** • In 8a., the value of i is equal to 1 not 22 as it displayed in the memory map in the first For Loop, second call. • in 8b., in the message board, ***copy string s[1] to t[ ]*** should be changed to ***copy string s[1] to t*** • In 8c., the cursor in the source code should be at the end of the scanf( ) when the message board displays "Let's assume that user enters 20" • In 9b., missing "}" at the end of For Loop • In 10d., the cursor in the source code should be at the end of the scanf( ) when waiting for the input for ***std.age*** and ***std.n***
Spelling	• In 3a., the heading left out the word "Basics" • in 5., ***An alphabetic letter*** should be changed to ***An alphabet character***,and ***a* = 3;*** should be changed to ***a = 3;*** • In 6., ***notaion*** should be change to ***notation*** • In 7., ***contain*** should be changed to ***contains, add 10 to whatever *iptr point to*** should be changed to ***add 10 to whatever iptr points to, meaning three objects*** should be changed to ***meaning third object, ckose*** should be changed to ***close*** • In 7a., the last line in the monitor output, ***FFB4*** should be changed to ***FFF4*** • in 8a., ***valu*** should be changed to ***value*** • in 9c., ***Grobal*** should be changed to ***Global***
Other	• Should provide an opportunity for the users to enter their own input. • Should include a voiced explanation option for the message board.

4.3.3 Implementation from Pilot Findings

The feedback from the subjects showed concern for the grammatical terms and concept correction. It was also suggested that the interface needed an update and some changes. The researcher reviewed the chapters and manipulated each animated example from the first one, *1: Flowchart*, in Chapter 1, to the last one, *10e: Reading a Binary File*, in Chapter 10.

Changes were implemented to the Animated Examples section as per most of the suggestions made by the subjects. This took two weeks. Some parts were left as they were because the suggested changes were substantial and were really only cosmetic changes to the interface of the animation such as bigger memory map window. Changes were made to content, accuracy and descriptions.

4.4 Conclusion

This chapter has described the development process and the design of DIVTIC, the DIVTIC set up website and the pilot study process. The necessary changes to the Animated Examples section in DIVTIC were completed and ready for the main study.

The aims of this study were to explore how students in introductory programming courses were aided by the use of interactive visual and Web-based instructional materials. The research questions covered two main areas:

1. How do students use the DIVTIC?
   - (1a) Which components of DIVTIC do students use and for how long?
   - (1b) What strategies do students use with DIVTIC?
   - (1c) What factors influence students' use of DIVTIC?
   - (1d) What attitudes do students generate towards DIVTIC?
2. To what extent does the dynamic interactive visualisation process implemented in DIVTIC influence learning outcomes?
   - (2a) How does the dynamic interactive visualisation process implemented in DIVTIC influence students' performance in programming?
   - (2b) How does use of the dynamic interactive visualisation process implemented in DIVTIC vary among students?
   - (2c) What levels and forms of cognitive engagement are evident among DIVTIC users?
   - (2d) What factors influence students' achievement with DIVTIC?

5.1 Research Study

In order to acquire the answers to the research questions, research methods were needed. Research methods can be classified in various ways. In educational research, however, one of the most common distinctions is between qualitative and quantitative research methods.

5.1.1 Quantitative Research

Quantitative research supports investigations that researchers can repeat to determine whether the same validity of the initial investigation results can be obtained by using the same procedures in another study (Bryman, 1989). Quantitative research emphasises the

testing of theory rather than generating and developing theory (Burns, 1994). It enables researchers to collect facts and study the relationship of a set of facts to another by using scientific measuring techniques to produce information in the form of numbers that can be quantified and summarised to produce a more generalisable picture of a problem (Bell, 1993). The most common quantitative research techniques include experimentation and surveys.

5.1.2 Qualitative Research

Since the early 1970s, there has been an increasing move in education towards qualitative research, which is concerned with gaining a deeper understanding of the individuals being studied (Bryman, 1989; Miles & Huberman, 1994). "Qualitative methods permit the evaluator to study selected issues in depth and detail. Approaching fieldwork without being constrained by predetermined categories of analysis contributes to the depth, openness, and detail of qualitative inquiry" (Patton, 1990, p. 13). It seeks to gain insight into human characteristics such as motivation, attitudes and behaviour in order to increase the understanding of a problem (Bell, 1993).

Qualitative methods provide the opportunity to immerse oneself into a situation in order to gain first-hand knowledge of the data collected (Miles & Huberman, 1994). Qualitative study is a descriptive form of research that seeks to explore "accurate and adequate descriptions of activities, objects, processes and persons" (Allison et al., 1996, p. 14).

Qualitative researchers can be found in many disciplines and fields, employing a variety of approaches, methods and techniques. Qualitative data is usually collected in the form of a written description based on observation, interviews, or documents which are normally not immediately accessible for analysis (Miles & Huberman, 1994). For example, with interview data, the researcher needs to record, transcribe or translate and correct before proceeding with analysis.

5.1.3 Triangulation

Both quantitative and qualitative methods have some advantages and disadvantages. A major difference between the two is that qualitative research is inductive and quantitative research is deductive. Quantitative methods seek to answer the question

'What is?' which stresses how social experience is created and given meaning, while qualitative methods seek to answer the question *'What if?'* which emphasises the measurement and analysis of the causal relationship between variables, not processes (Allison et al., 1996; Denzin & Lincoln, 1998; Moore, 2000). In qualitative research, a hypothesis is not needed to begin the research. However, quantitative research requires a hypothesis before any research can begin.

In terms of data collection, Trochim (2000) argues that there is little difference between qualitative and quantitative data because all qualitative data is based on qualitative judgment and can be coded quantitatively. Allison et al. (1996) supports this argument:

> ... human attributes such as intelligence, happiness and personality characteristics as well as people's values and opinions, including such as those concerned with assessments of beauty and intensity of religiosity, are also variables and so, with more or less degree of precision, are able to be measured quantitatively. (p 14)

Although most researchers do adopt either a quantitative or a qualitative research method, some researchers have suggested combining the two methods in the one study; an approach that Patton (1990) and Trochim (2000) believe is valuable to almost every applied social research project.

Overall, "Because qualitative and quantitative methods involve differing strengths and weaknesses, they constitute alternative, but not mutually exclusive, strategies for research. Both qualitative and quantitative data can be collected in the same study" (Patton, 1990, p. 14). One can be used in conjunction with the other if the researcher plans carefully and uses each method in a thoughtful manner (Snyder, 1995). Moreover, it is suggested that to overcome the weaknesses of each method, a combination of data collection techniques should be used from both qualitative and quantitative paradigms (Erzberger & Prrein, 1997). This combination of both qualitative and quantitative methods is a form of triangulation which Patton (1990) describes as data collected from both paradigms:

> Triangulation of qualitative and quantitative data is a form of comparative analysis [...] This means comparing and cross-checking the consistency of information derived at different times and by different means within qualitative methods. (pp. 466-467)

Creswell (1994) describes three models of combined designs:

1. Two-phase design: The researcher conducts a study in one paradigm followed by another. The two paradigms are clearly separated and presented, but the reader may not recognise the connection between the two paradigms;
2. Dominant-less dominant design: The researcher presents a study in one dominant paradigm and uses a small component drawn from the alternative paradigm; and
3. Mixed-methodology design: The researcher presents both paradigms at all stages.

Previous researchers, for example Boatwright and Slate (2000), have used a method of two-phase design to investigate work ethics as defined by the Georgia Department of Technical and Adult Education. They used a qualitative method in the first stage followed by a quantitative method in the second stage. Sale, Lohfeld, and Brazil (2002), also point out that there are many studies in health care research that combine both qualitative and quantitative paradigms. The research presented in this thesis combines a number of techniques for data collection by using the mixed-methodology design as described by Creswell (1994).

For this study, a mixed mode of quantitative and qualitative was chosen. A quantitative design was chosen to determine if there would be improved performance among students and a qualitative design was to be used to explore the impact of DIVTIC and to establish causal relationships between its use and the performance of the students.

The quantitative method was quasi-experimental, involving an experiment with a control group and an experimental group. Both control and experimental groups were treated in the same manner except that the experimental group was provided with access to using DIVTIC as a supplementary tool. However, the nature of educational settings is such that control of variables is very difficult to guarantee.

The qualitative method used was an ethnographic approach involving observation and interviews with students and their tutors in the use of DIVTIC.

5.1.4 Designing Choice of Research Strategies

To minimise the distinction between qualitative and quantitative research paradigms, Bryman (1988) suggests selecting techniques for data collection suited to specific research questions. According to Sproull (1988), there are four main techniques for collecting qualitative and quantitative data:

1. interviewing;
2. instrument administration;
3. observation; and
4. examination of documents, materials and artifacts.

Elements from both paradigms were suitable for this research. Appropriate data collection strategies included the observation of students, the interviewing of students and tutors, document examination and instrument administration e.g., questionnaires, laboratory tests, midterm examination, etc. Data analysis included the use of scientific measurement software for statistical analysis as well as the interpretive analysis of screen recordings. The variety of analytical techniques selected ensured that the data fully and accurately answered the research questions. The following sections describe the basic characteristics upon which each technique was chosen.

5.1.4.1 Interviewing

Interview techniques involve systematically collecting verbal information about the interviewee's opinions, attitudes, values, beliefs or behaviors (Sproull, 1988). They have been used in a wide variety of research projects and can be unstructured, semi-structured, or structured. The main benefit of using the interview technique is that interviewees can express their feelings and opinions in their own terms (Patton, 1990). Furthermore, this method provides in-depth information. Sewell (n.d.) writes that the interview technique is most useful for:

1. evaluating programs that are aimed at individualised outcomes;
2. capturing and describing program processes;
3. exploring individual differences between participants' experiences and outcomes;
4. evaluating participants' evolving understanding of a program; and
5. documenting variations in program implementation at different sites.

Patton (1990) describes three basic approaches for data collection through open-ended interviews:

1. the informal conversational interview;
2. the general interview guide approach; and
3. the standardised open-ended interview.

Each approach has its strengths and weaknesses as shown Table 5.1.

Table 5.1: Variation in interview instrumentation (Adapted from Patton, 1990, pp. 288-289)

Type of interview	Strengths	Weaknesses
Informal conversational interview	▪ Increases the salience and relevance of questions ▪ Interviews are built on and emerge from observations ▪ The interview can be matched to individuals and circumstances	▪ Different information collected from different people with different questions ▪ Less systematic and comprehensive if certain questions do not arise "naturally" ▪ Data organisation and analysis can be quite difficult
General interview guide approach	▪ The outline increases the comprehensiveness of the data and makes data collection somewhat systematic for each respondent ▪ Logical gaps in data can be anticipated and closed ▪ Interviews remain fairly conversational and situational	▪ Important and salient topics may be inadvertently omitted ▪ Interviewer flexibility in sequencing and wording questions can result in substantially different responses from different perspectives, thus reducing the comparability of responses
Standardized open-ended interview	▪ Respondents answer the same questions, thus increasing comparability of responses ▪ Data are complete for each person on the topics addressed in the interview ▪ Reduces interviewer effects and bias when several interviewers are used ▪ Permits evaluation users to see and review the instrumentation used in the evaluation ▪ Facilitates organization and analysis of the data	▪ Little flexibility in relating the interview to particular individuals and circumstances ▪ Standardized wording of questions may constrain and limit naturalness and relevance of questions and answers

A standardised open-ended interview format was chosen for this study because this technique facilitates the collection, organisation and analysis of completed data in a limited period of time. Patton (1990) writes that there are basically six kinds of questions that can be asked of people:

1. experience/behaviour questions;
2. opinion/values questions;
3. feeling questions;
4. knowledge questions;
5. sensory questions; and
6. background/demographic questions.

The standardised open-ended interview format would allow this study to explore the students' perception of ease of use, look and feel, contribution to learning, and conceptual understanding.

5.1.4.2 Instrument and Questionnaires

Sproull (1988), defines the term instrument and gives the following examples:

> An instrument is whatever device is used to measure variables. Instruments can range from written or oral materials to physical devices. Examples of instruments include: (1) questionnaires (e.g., asking opinions of recent mergers), (2) rating scale (e.g., rating major corporations on the social goals), (3) skill test (e.g., a typing test), (4) checklists (e.g., checking cities which have a high 'quality of life') and (5) materials created by Ss (e.g., Ss designing parts for a computer). (p. 175)

This study sought to use three instruments: a questionnaire, tasks, and experimentation.

- **Questionnaires:** Questionnaires can be divided into two broad types: structured or unstructured. As Trochim (2000, para. 2) states, "Structured formats help the respondent to respond more easily and help the researcher to accumulate and summarize responses more efficiently. But, they can also constrain the respondent and limit the researcher's ability to understand what the respondent really means." On the other hand, unstructured formats give respondents more opportunity to express their thoughts in their own words. This also applies to open-ended questionnaires where subjects are able to add further comments in their own words. In this study, a structured format was used to facilitate the collection of quantitative data involving students' attitude (see Section 5.4.4.1).
- **Tasks:** A task is a problem which needs to be clarified with a solution. A task is a very useful instrument to direct the students to creating consistent activities. It encourages students to be active learners. In this study weekly tasks were developed and given to the students to complete before testing their answers using DIVTIC. This was to ensure that the students actually used DIVTIC to test their answers. Data was collected from the results of these tasks (see Section 5.4.4.7).
- **Experimentation:** Experimentation deals with measurable phenomena whereby conclusions are drawn to establish cause and effect relationships in a controlled situation (Bell, 1993). Turney and Robb (1971) suggest that the

most appropriate method for the investigation of problems in education is experimental research by which one or more factors are systematically varied in order to determine what effects each variation produces. This technique was used in this study to explore students' achievement by:

(a) providing some assessment instruments including an initial laboratory test, a midterm examination, a second laboratory test and the final examination. The test results constituted as data to be calculated and analysed (see Section 5.4.3.1 and 5.4.3.2);

(b) using a cgi script to record and keep track of students' log-in time. This record helped to determine whether the time spent using DIVTIC made any significant difference to students' outcomes (see Section 5.4.4.6); and

(c) checking and analysing on the use of C WebBoard (Section 4.1.2.5). All messages were kept in a database on the server. The data from this source recorded any collaboration that occurred in this setting.

5.1.4.3 Observation

The observation technique is the systematic recording of a subject's behavior patterns without questioning or communicating with them. Sproull (1988) defines this as "A data collection method in which a person (usually trained) observes Ss or phenomena and records information about characteristics of the phenomena" (p. 166). There are two main types of observation: participant and non-participant observation, which can be part of either quantitative or qualitative research (Bell, 1993). Observation enables first-hand knowledge of the context in which events occur and allows the researcher to see things that the participants themselves are not aware of or are unwilling to discuss (Patton, 1990). Patton (1990) notes that:

> What people say is a major source of qualitative data, whether what they say is obtained verbally through an interview or in written form through document analysis or survey responses. There are limitations, however, to how much can be learned from what people say. To understand fully the complexities of many situations, direct participation in and observation of the phenomenon of interest may be the best research method. (p. 25)

This notion is supported by Marshall and Rossman (1999) who claim that "Observation is a fundamental and highly important method in all qualitative inquiry: It is used to

discover complex interactions in natural social settings" (p. 107). The observation method was considered to be a useful data collection strategy for this study because only this type of data would help to confirm students' opinions. Observation was used in three different ways in this study:

1. The Researcher observation form was used to record any problems during the laboratory session regarding hardware, software, the network, etc. In this way, the researcher could either resolve the problem or ask a technician to do so. This ensured that a minimum of difficulties was encountered within the learning environment.
2. The Tutor observation form was used to record students' expressions and behaviour when asking questions about DIVTIC. Tutors were asked to note any problems that occurred during the experimental session relating to hardware, software, the network, etc. Tutors responded to three questions: What questions did the students ask when using DIVTIC? How easy was DIVTIC for the students? and What problems did you face this week relating to DIVTIC with hardware, software, the network, etc?
3. The screen video capture software was used to record students' use of DIVTIC. This enabled the researcher to explore how students used DIVTIC by looking at the recorded video to see what extent it influenced students' higher-order thinking, confidence, and motivation.

5.1.4.4 Examination of Documents, Materials and Artifacts

This is another data collection method which uses existing data. In this study, this method ensured that both the control and experimental groups were perfectly matched. The Grade Point Average (GPA) of each student, from the previous trimester, was used to match the two groups. This is described in detail in section 5.3.2.

5.2 Data Matrix

The data for this study was collected by using the following 10 strategies:

1. DIVTIC self-administered evaluation questionnaires;
2. Subject semi-structured interviews;
3. Tutor observation;
4. Researcher observation forms;

5. Tutor semi-structured interviews;
6. First and second laboratory tests plus midterm and final examinations;
7. Screen recordings;
8. DIVTIC weekly tasks;
9. DIVTIC log in records; and
10. C WebBoard

These were planned to be used in the following ways:

Table 5.2: Data matrix

Research Question	Method	Data Collection	Data Analysis
1. How do students use DIVTIC?			
(1a) Which component of DIVTIC do students use and for how long?	(1a) A, G, and I	(1a, 1b) Collect subjects' attitudes and performances data from A (weeks 6 and 10) and B, G, and I (weekly)	(1a, 1b, 1d) Descriptive analysis
(1b) What strategies do students use with DIVTIC?	(1b) A, B, and G		
(1c) What factors influence students' use of DIVTIC?	(1c) A, B, C, D, E, and G	(1c, 1d) Collect data relating to the effects of using DIVTIC from A (week 6) and E (weeks 7 and 11) and B, C, D, and G (weekly)	(1c) Qualitative analysis to identify patterns in improvement towards DIVTIC
(1d) What attitudes did students generate towards DIVTIC?	(1d) A, B, C, and E		
2. To what extent does the dynamic interactive visualisation process implemented in DIVTIC influence learning outcomes?			(2a) Mean comparisons to investigate differences between the experimental and control groups
(2a) How does the dynamic interactive visualisation process implemented in DIVTIC influence students' performance in programming?	(2a) F	(2a, 2b) Collect lab test 1, midterm, lab test 2, and final scores from F (weeks 7, 7, 11, and 13, respectively)	(2b) Mean comparisons to investigate differences between each level of students' GPA in the experimental groups
(2b) How does use of the dynamic interactive visualisation process implemented in DIVTIC vary among students?	(2b) F		(2c) Descriptive analysis
(2c) What levels and forms of cognitive engagement are evident among DIVTIC users?	(2c) C, G, H, and I	(2c) Collect data from C, G, H, and I (weekly)	(2d) Inferential analysis to investigate relationships between achievement and time spent in DIVTIC, and computer experiences.
(2d) What factors influence students' achievement with DIVTIC?	(2d) A and I	(2d) Collect data from A (weeks 6 and 12) and I (weekly)	

NOTE:
A: DIVTIC self-administered evaluation questionnaires
B: Subject semi-structured interviews
C: Tutors' observations
D: Researcher's observations
E: Tutor semi-structured interviews
F: Lab tests 1 and 2, Midterm and final examinations
G: Screen recording
H: DIVTIC weekly task
I: DIVTIC log file

5.3 The Study

The research for this study was conducted in Thailand at Suranaree University of Technology (SUT), during the second trimester of 2001 from September to December. There are 13 weeks in each trimester. In their first trimester, students are required to take five basic courses and one associated laboratory: Chemistry I (4 credits), Chemistry Lab. (1 credit), Calculus I (4 credits), Logical Thinking (3 credits), IT I (3 Credits), and English I (3 credits). There are 18 credits all together in the first trimester. The IT I (Information Technology 1) covered a small part of computer programming in C at the end of the course. Overall, the students who participated had very little knowledge of computer programming.

To be able to graduate in four years, the subjects were required to enrol in the following second trimester units: Chemistry II (4 credits), Chemistry Lab II (1 credit), Calculus II (4 credits), Physic I (4 credits), Physic Lab I (1 credit), and Computer Programming in C (3 credits). This totalled 17 credits all together. Therefore, students had to study hard to complete the requirement.

5.3.1 Classroom Process

Computer Programming 408101, is a basic computer programming course which is a requirement for all engineering students. It is only offered in the second trimester although the School of Computer Engineering may also offer it in the third trimester. However, to be able to graduate in four years, students need to enrol in this course in the second trimester of their first year since it is a prerequisite for later courses. This course teaches the basic concepts of object-based programming using C. Students who want to be accepted into the School of Computer Engineering have to receive a C grade or above. The objectives of the course are to help students understand the common components and principles of a programming language and to be able to solve logical problems by using the C computer programming language.

In the period when this study was conducted, in 2001, there were approximately 500 undergraduate engineering students enrolled in Computer Programming 408101, of whom 100 took part in the study. The laboratory component of the course was divided into 10 sessions. The students registered into one of these 10 sessions on a first-come first-served basis. There were 50 students and two tutors in each session. There were

four computer laboratories, each with 60 networked personal computer systems. The laboratory sessions were divided into three time-slots as follows:

- Time-slot 1: Tuesday 4.00-6.00 p.m.;
- Time-slot 2: Wednesday 4.00-6.00 p.m.; and,
- Time-slot 3: Wednesday 6.00-8.00 p.m.

The first two laboratory sessions, 1 and 2, were in Time-slot 1. Sessions 3, 4, 5, and 6 were in Time-slot 2 and the rest, sessions 7, 8, 9, and 10 were in Time-slot 3

In order to have the same tutors for both the control and experimental groups, the study was designed to use session 6 as a control group and session 10 as an experimental group. Both tutors were experts in C programming. One of them had tutored in C for many years and the other was an instructor who had previously taught this class and who had written a Thai version of a C Programming textbook.

The study was conducted in normal classes, with the researcher providing guidance to the participating teachers and acting as an observer in the classes when the experiment was conducted. The students were informed by their teachers of the study and the role of the observer.

5.3.2 Subjects' Setting

The Grade Point Average (GPA) of students in each session for the first trimester for both Time-slot 2 and Time-slot 3, was used to match up two groups. The GPA record was obtained from the Centre for Educational Services, SUT.

Convenience sampling was used to select subjects by manually matching the GPA between Time-slot 2 and Time-slot 3 in choosing sample groups. Convenience sampling is defined by Sproull (1988, p. 117) as "A nonrandom sampling method in which the researcher uses some convenient group or individuals as the sample." In this study, for example, session three to six in the Time-slot 2 were comprised of up to 50 students ranging from low to high GPA who were then assigned into session six and called the control group (Group C). In addition, sessions 7 to 10 in Time-slot 3 were both selected with up to 50 students in the same range as the control group from low to high GPA and were assigned into session 10 and called the experimental group (Group E).

Each group was divided into three different levels according to their GPA: low (1.00 to 1.72); average (1.78 to 2.22); and high (2.28 to 3.36). Group C, for example, included C1, C2, and C3 which referred to low, average, and high GPA respectively, and Group E was comprised of E1, E2, and E3 which also referred to low, average, and high GPA respectively. Dividing the students by their GPA into three levels helped the researcher determine the different achievements of each level. Finally, both groups were perfectly matched by GPA as well as by gender. Each group comprised a matched set of students as shown in Table 5.3.

Table 5.3: Matching number of students in both groups based on gender

Gender	GPA 1.00 – 1.72		GPA 1.78 – 2.22		GPA 2.25 – 3.36	
	C1	E1	C2	E2	C3	E3
Male	8	8	9	8	8	9
Female	8	8	8	9	9	8

A quasi-experimental design was used in this study since the subjects were not randomised. The experimental group used DIVTIC throughout the trimester while the control group did not. Table 5.4 shows that both the experimental and control groups were treated in the same manner except for use of the DIVTIC system.

Table 5.4: Quasi-experimental design without pretest in both groups

Group	DIVTIC	Lab Test 1	Midterm	Lab Test 2	Final
Experimental	✓	✓	✓	✓	✓
Control	✗	✓	✓	✓	✓

5.4 Resources

There were four sets of resources required for this study: hardware, software, instructor's printed media, and researcher's printed media and file. The hardware and software were already prepared and set by the Center for Computer Services (CCS) at SUT. However, since this was an important research resource, it is described in detail below.

5.4.1 Hardware

The hardware in the laboratory had the following specifications:

- Desktop computer: Celeron 500Mhz, RAM 128 MB and HD: 13GB.
- Operating System: Windows 98 se Thai version.
- Connection: Using HUB to gateway: 203.100.0.254.

5.4.2 Software

The software included:

- Internet Explorer 5.5: A browser software program developed by Microsoft Corp. downloadable at http://www.microsoft.com/windows/ie/default.asp
- Borland Turbo C 3.0: The C compiler software from Borland Software Corporation. For more information go to http://www.borland.com
- Camtasia 3.0.0 (Trial version): A screen video capture software for screen recording from TechSmith Corporation, downloadable at http://www.techsmith.com
- Acrobat Reader 4.0: A freely distributed program from Adobe Systems, Inc. used to view PDF files created with Adobe Acrobat or other programs, downloadable at http://www.adobe.com
- Winzip 6.2: A program that enables the archiving and compressing of files available from WinZip Computing, Inc., downloadable at http://www.winzip.com

5.4.3 Instructor's Printed Media

The four printed handouts developed for the study that were given to all students by the instructor at various stages are described as follows.

5.4.3.1 Laboratory Test

There were two laboratory tests during the entire course. Each test was comprised of two short questions asking students to write C source code to solve given problems. Each laboratory test, taken in the laboratory session, lasted 1 hour and 45 minutes and was worth 10 points. The subjects had an opportunity to test their source code via the C compiler before handing it to the tutors.

5.4.3.2 Midterm and Final Examinations

Both midterm and final examinations were open-book tests and comprised 60 multiple-choice questions. Each examination was 2 hours in duration. The midterm examination was worth 30 points while the final examination was worth 50 points.

5.4.3.3 Instructor Weekly Problem

The instructor created 10 weekly problems relating to the course outline and gave them to all subjects. Each weekly problem contained two small problems asking the subjects to solve them by writing C source code to test via the C compiler in the laboratory. There were no marks given for the weekly problems. They were designed to encourage the subjects to test their own understanding and ability. The instructor also posted the solution to each weekly problem on his website 2 weeks after it was given to the subjects.

5.4.3.4 Lecture Notes

The instructor gave out his lecture notes in two forms: in hard copy and as Power Point slides in a downloadable zip file format. The subjects were able to go to the instructor's website to download and print the lecture notes before the lecture. The subjects therefore knew the key words and topics covered in that particular week and they could annotate the downloaded handout.

5.4.4 Researcher's Printed Media and File

The handouts given to the subjects were created by the researcher and they are described in the following sections.

5.4.4.1 DIVTIC Self-administered Evaluation Questionnaire Form

The DIVTIC self-administered questionnaire contained three parts: the questionnaire, the checklist, and the open-ended question. The questionnaire was comprised of 33 short questions using a five-point Likert rating scale. It was divided into 11 patterns which examined the subjects' experiences such as their higher-order thinking, confidence, motivation, etc. The three checklist questions examined the frequency and types of strategies used, and when the subjects used DIVTIC. The open-ended question

gave room for students to elaborate on any problems using DIVTIC. Figures 5.1 and 5.2 demonstrate the structure of the DIVTIC self-administered questionnaire

Part 1: Please indicate how much you disagree or agree with each of the following statements.

Note: 1 = Strongly Disgree
2 = Disgree
3 = Not Applicable
4 = Agree
5 = Strongly Agree

	QUESTION	1 Strongly Disagree	2 Disagree	3 Not Applicable	4 Agree	5 Strongly Agree
Higher-order Thinking	When I am watching the animation, I stop it from time to time to reflect on what I am trying to get out of it.					
	Using Animated Examples help me to think logically duringthe animation process.					
	I always discuss with my peers about the animation running process.					
Confidence	Using Animated Examples increase my confidence in learning programming.					
	Using Animated Examples, I believe that I can solve more complicated tasks.					
	Using Animated Examples, I feel that I can help other peers in solving a given problem.					
Encouragement	Using Animated Examples, I feel that I pay more attention in programming class.					
	Using Animated Examples encourage me in programming more efficiently.					
	Using Animated Examples, I feel that programming is not too difficult to learn.					
User Friendliness	The interface of Animated Examples is pleasant.					
	Animated Examples are an easy-to-use tool.					
	Animated Examples are easy to navigate.					
Perceived Enjoyment	I enjoy using Animated Examples.					
	Animated Examples entertain me in learning programming.					
	I feel comfortable by using Animated Examples.					
Perceived Educational Value	I am pleased to have Animated Examples as assistance.					
	Animated Examples are a useful tool in learning how to program in C.					
	Animated Examples material is challenging.					

Figure 5.1: DIVTIC Self-administered evaluation questionnaire Part 1

Part 2: Please tick all that apply.

1. How many times have you used the following parts of DIVTIC in the last two weeks?

Topic	0	1 - 3	4 - 6	7 - 10	> 10
Syllabus/Lecture Note					
Computer Structure					
Animated Examples					
"C" Compiler					
"C" WebBoard					
Self-evaluation					
FAQ Pool					
"C" References & Links					

2. How do you use "Animated Example"?

- ☐ I just watch the animation withoutinteracting with it.
- ☐ I press **STOP/PLAY** button to think along what is going on.
- ☐ I repeat the animation to make it clear of how the program executes.
- ☐ I press Backward/Forward to see the animation.
- ☐ I stop the animation and discuss with my peers.
- ☐ I go to test my own code right after watching the animation to compare the output
- ☐ Others:________________

3. When do you normally use DIVTIC?

- ☐ When I am in a laboratory session.
- ☐ When I have free time.
- ☐ When I do assignments.
- ☐ Before examinations.
- ☐ When I face with programming problems.
- ☐ When I am with friends.
- ☐ Others:________________

Figure 5.2: DIVTIC Self-administered evaluation questionnaire Part 2

The data from this form enabled the researcher to answer some parts of the first research questions 1b, 1c, and 1d, and the second set of research questions 2c, and 2d.

5.4.4.2 Tutor Observation Form

The Tutor observation form was an open-ended questionnaire for tutors to write a weekly report based on their observations of the use of DIVTIC. There were three open-ended questions to describe the subjects' attitudes toward the use of DIVTIC and any general problems which occurred during each laboratory session including software, hardware and the network. The questions were as follows:

- Question 1: What questions did students ask when using DIVTIC?
- Question 2: How easy was DIVTIC for students to use?
- Question 3: What problems did you face in this week in relation to DIVTIC including hardware, software, the network, etc.

Once again, the data from this observation enabled the researcher to explore and answer the first research questions 1c and 1d, which examined the subjects' attitudes toward the use of DIVTIC with regard to its user-friendliness, useability, level of enjoyment and other factors.

5.4.4.3 Researcher Observation Form

The researcher took notes on a weekly basis during the observation sessions to investigate any unexpected problems to do with software, hardware, networking, and other things. The observation was aimed at discerning any problems that occurred which influenced the subjects' use of DIVTIC.

5.4.4.4 Subject Semi-structured Interview Form

In this study all subjects were asked the same questions from an open-ended question interview form. This makes the process of data analysis easier as each subject's response to a particular question is easy to locate, and similar responses are easy to group together (Patton, 1990). The form contained five open-ended questions relating to the subjects' attitudes and the use of DIVTIC. The interview was undertaken during the laboratory session in a quiet corner of the room. The Following table (Table 5.5) shows

the questions and their kind described by Patton (1990) to explore the students perception.

Table 5.5: Subject semi-structured interview and Tutor semi-structured interview forms

Question		Kind
1.	Feedback on how many problems the subjects experienced.	Experience/Behaviour
2.	Did they finish the weekly task? If not, why not?	Experience/Behaviour
3.	Some questions about the specific problems, e.g. asking about the algorithm to see if the students actually learned about it	Knowledge
4.	Which part of the animation helps you learn the most, and why?	Sensory
5.	What should be improved to make DIVTIC a better tool?	Opinion/Values

The findings from this interview were an invaluable source of information since the interviewees had the opportunity to express their views, ideas, feelings, attitudes, etc. in their own words. The interviews were taped and transcribed for analysis.

5.4.4.5 Tutor Semi-structured Interview Form

The Tutor semi-structured interview form contained five open-ended questions which examined tutor attitudes and suggested improvements to DIVTIC. The interviews were undertaken during laboratory tests 1 and 2, outside the laboratory. The following table (Table 5.6) shows each question and its kind as described by Patton (1990).

Table 5.6: Tutor semi-structured interview forms

Question		Kind
1.	What do you think about Animated Examples e.g., interface, usability, clarity, user-friendliness, and value?	Opinion/Values
2.	As a tutor, do you like Animated Examples, how and why?	Felling
3.	As a student, do you like Animated Examples, how and why?	Opinion
4.	What other features do you think Animated Examples should have?	Opinion/Values
5.	Do you have any other comments about Animated Examples?	Opinion/Values

The findings from this interview enabled the researcher to explore how to improve the useability and feasibility of DIVTIC for further implementation and to find out some factors that could influence students' use of DIVTIC. This provided invaluable feedback since both interviewees were experts in C programming. These interviews were also taped and transcribed for analysis.

5.4.4.6 DIVTIC Log File

A cgi script was used to record the subjects' log-in time. A record was kept on a file in the database system whether the log in process was successful or not. The log file format recorded information including the IP address, date and time, the subject's identification and the pages the subject accessed. Each time a subject accessed another page, the cgi script would record that activity. Each Log file format was then converted to text file format and transferred into an Excel file for the convenience of calculating the log-in time as shown in Figure 5.3.

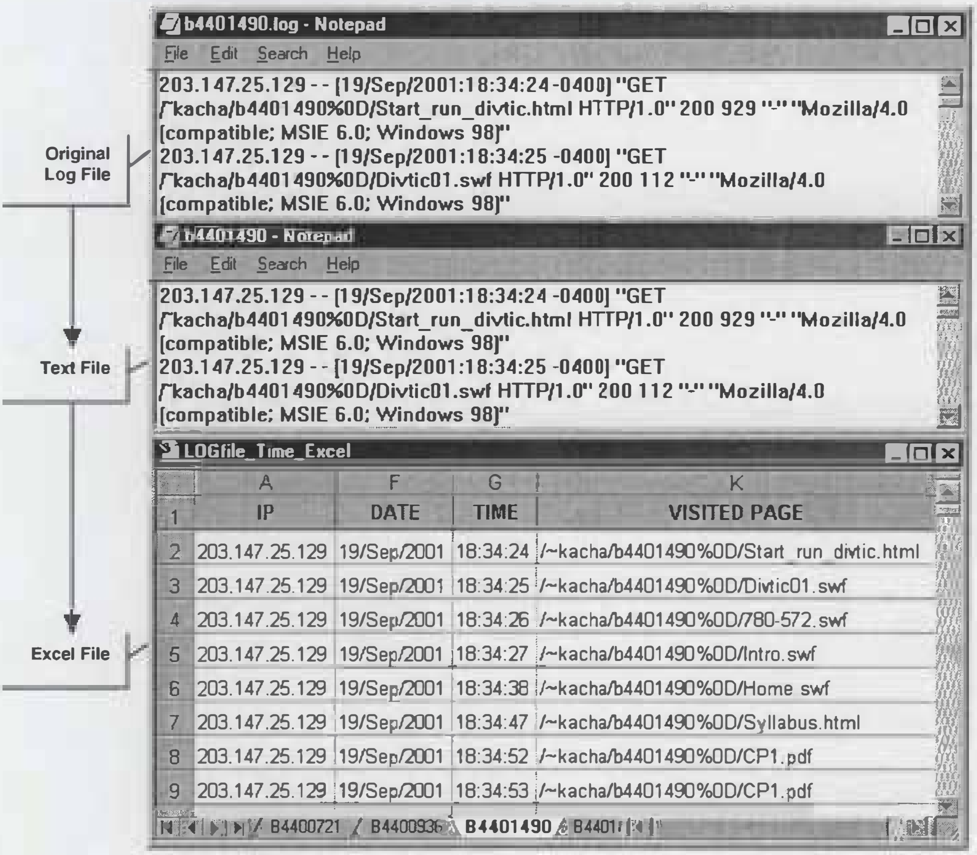


Figure 5.3: Converting from log file to text file and then to Excel file

5.4.4.7 DIVTIC Weekly Tasks

The subjects were required to complete a weekly task sheet by filling in the answers before running DIVTIC. The subjects then needed to run an associated animated example in DIVTIC and write down the answer. They then compared both answers. If the answer to the task was incorrect, they were asked to write some short messages explaining why they had made a mistake, e.g., they did not understand the question, forgot to increase the counter, and so on. These messages were collected and are

discussed in Chapter 6. The weekly task was comprised of two short questions. Each question included four parts:

1. Source Code: A given source code for the subjects to go through;
2. Your Answer: A space provided for the subjects to fill in the answer;
3. DIVTIC System: A duplicate version of 'Your Answer' which was provided for the subjects to fill in the answer after watching the associated animation. In this case, it was called "Input/Output using scanf( ) and printf( )"; and,
4. Note: A space for the subjects to write down why they did it incorrectly.

A sample of problem 1 in weekly task 2 is shown as follow:

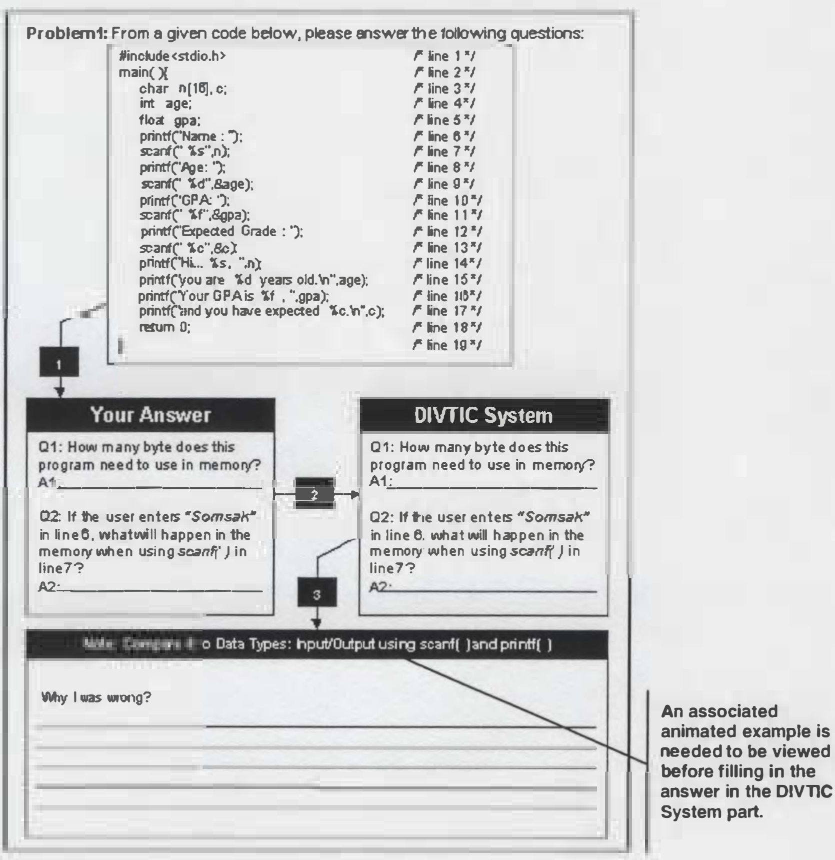


Figure 5.4: An example of problem 1 in weekly task 2

5.4.4.8 Screen Recording

Screen video capture software was used to record the activities of three voluntary subjects using DIVTIC for 30 minutes a week. This screen recording enabled the researcher to see which part the subjects used the most and how they used them. For

example, whether the subjects simply let the animation go through the process without interacting with it or whether they interacted with the animation by clicking the buttons, e.g. *Play, Stop, Step Forward*, etc., on the control menu as they thought through the process. If the subjects pressed the *Go To The End* button at the beginning of the animation process, it was revealed that they were not using DIVTIC to enhance their learning and understanding. A still sample of a screen recording is given in Figure 5.5.

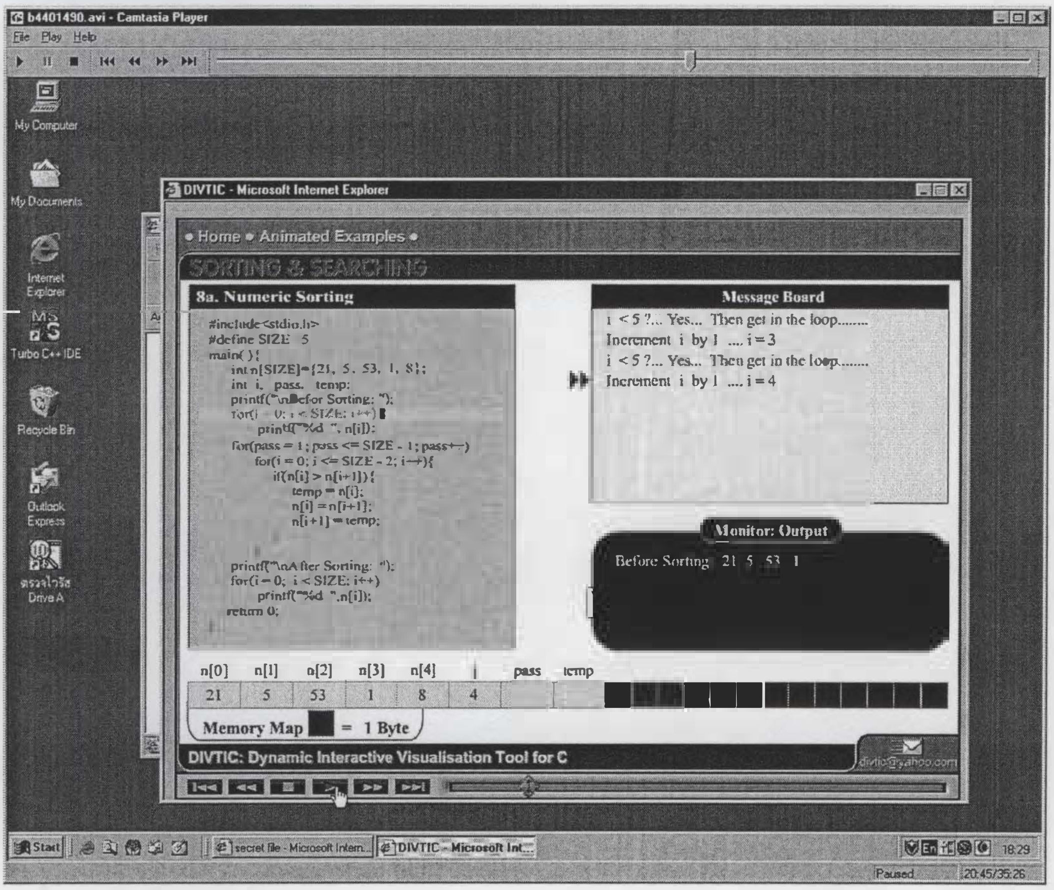


Figure 5.5: A sample of screen recording

5.5 Procedure

The data collection was conducted at SUT in the second trimester of 2001 starting 17 September and ending 16 December. The conventional teaching practice of "Computer Programming 408101" included a two-hour lecture in a lecture theatre and a two-hour laboratory session which was scheduled after the lecture.

The first lecture was given on Tuesday, 18 September 2001, in a big lecture theatre holding approximately 450 students. The researcher asked the instructor to announce in class that the students who registered for laboratory sessions 3 to 10 needed to check for

their names on the list posted on each laboratory door. Most students stayed in the same session for which they had registered. Some changed to a different session but stayed in the same time-slot. The details of each week are described as follows.

Week 1: Flowchart

In the first week of class the subjects in the experimental group were informed about the study. The researcher introduced himself and explained the purpose and importance of his research to the students and tutors. The researcher then gave consent forms to each of the subjects, asking them to sign it if they agreed to be a part of the research. If anyone did not want to take part, they were welcome to go into another session without penalty.

All the selected students agreed to be a part of the research and signed the consent form and returned it to the researcher. The researcher distributed passwords, demonstrated how to log in, explained how to navigate in the DIVTIC system and highlighted the various features available in it. The remaining time was used to complete the weekly problem given by the instructor. The researcher's weekly task was not given to the subjects in the first week because of insufficient time. The task for week one, the Flowchart, was given in the following week.

Week 2: Data Types & Input/Output

The weekly task for week one, the Flowchart, and week two, Data Types & Input/Output, were given to the subjects at the beginning of the session. The subjects were asked to complete them within 45 minutes and use the remaining time to complete the instructor's weekly problem. However, the subjects took up to 1 hour 15 minutes to complete both tasks. It took more time than expected because it was the students' first exposure to the task and they were unfamiliar with DIVTIC. The tutors collected the two tasks and handed them to the researcher.

The researcher asked for three volunteers to be interviewed using the Subject semi-structured interview form. Each interview was tape-recorded and took about 5 minutes. The Tutor observation form was also given to the tutors to complete and return to the researcher at the end of the session. The researcher also used the Researcher observation form to record his observations.

Week 3: Operators to Week 6: Arrays

From weeks 3 through 6 the procedures followed along the same lines:

- At the beginning of each laboratory session, a weekly task was given to the students who were allowed approximately 30 to 40 minutes to complete the task sheet, depending on its level of difficulty. At the same time, the screen video capture software was used to record the usage of DIVTIC by three volunteers. Each recording lasted 30 minutes.
- The tutors collected the weekly task, handed it to the researcher and let the subjects do the instructor's weekly problem.
- Three volunteers were interviewed and each interview took about 5 minutes.
- Tutors completed the Tutor observation form and handed it to the researcher.
- The researcher also completed the Researcher observation form.

The only extra instrument given to the subjects in week 6 was the Self-administered questionnaire form. It took about 10 minutes to complete the form.

Week 7: Review for Midterm

Two tests were given in this week: laboratory test 1 and the midterm test. Only one instrument, the Tutor semi-structured interview form was used.

Week 8: Pointer to week 10: Structure

The activities in weeks 8 to 10 were the same as in weeks 3 to 6. The Self-administered questionnaire form was given in week 10.

Week 11: Data Files

Laboratory test 2 was given in this week. Only one instrument, the Tutor semi-structured interview form was used.

Week 12: Review for Final Examination

There were four instruments used in this week including the Tutor observation form, the Researcher observation form, the weekly task and the screen video capture software.

Week 13: Final Week

In this final week the final examination was given. There was no other activity during this week.

Overall both the control and experimental groups had the same instruments provided by instructor including laboratory test 1, the midterm examination, laboratory test 2, and the final examination. The extra instruments for the experimental group, used by the researcher, are described in the following table (Table 5.7).

Table 5.7: Extra instruments using in the experimental group from weeks 1 to 12

Instrument	Week											
	1	2	3	4	5	6	7	8	9	10	11	12
Weekly Task		✔	✔	✔	✔	✔		✔	✔	✔		✔
Researcher observation		✔	✔	✔	✔	✔		✔	✔	✔		✔
Tutor observation		✔	✔	✔	✔	✔		✔	✔	✔		✔
Self-administered Questionnaire form						✔				✔		
Subject semi-structured interview form		✔	✔	✔	✔	✔		✔	✔	✔		
Tutor semi-structured interview form							✔				✔	
DIVTIC Log file		✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	
Screen Video Capture software			✔	✔	✔	✔		✔	✔	✔		✔

5.6 Data Gathering

The following procedures for collecting the data were enforced:

- **The midterm and final examinations:** Midterm and final examinations were taken in weeks 7 and 13, each worth 30 and 50 points, respectively.
- **Laboratory test 1 and 2:** Laboratory test one and two were taken in weeks 7 and 11, respectively, and were each worth 10 points.
- **The DIVTIC self-administered evaluation questionnaire form:** The self-administered evaluation questionnaire form was given to the subjects in weeks 6 and 10 at the beginning of the session. Then, tutors collected and gave it to the researcher at the end of that session.
- **The Tutor observation form:** The tutors were asked to complete the tutor observation form that included three weekly open-ended questions starting from week 2 onwards then returning it to the researcher.
- **Researcher observation form:** The researcher used an observation form to record the use of DIVTIC by the subjects and the setting up and running of the DIVTIC system starting weekly from week 2 onwards.

- **The Subject semi-structured interview form:** The Subject semi-structured interview forms were used by the researcher to interview three volunteers individually from the experimental group during weekly laboratory session from week 2 until the end except for the weeks 7 and 11 which were the laboratory test weeks. Each interview took about 5 minutes and was tape-recorded. To make it easy to transcribe the tape-recording, the researcher changed the format to record from an analog signal on the tape to a wave file format (*.wav) which was an audio file format created by Microsoft to be used primarily on personal computers (*Wave File*, 2002). However, each file was still too large and those files needed to be compressed. The MP3 (MPEG-1 Audio Layer-3) format was chosen as a final version. Each wave file format was compressed by a factor of 12 in size without losing sound quality (Watson, 2002).
- **The Tutor semi-structured interview form:** A Tutor semi-structured interview form was used to interview both the tutor and instructor in weeks 7 and 11. Each interview took about 10 minutes and was tape-recorded.
- **DIVTIC log file:** When the subjects logged into DIVTIC, the DIVTIC system read and wrote each subject's log-in time in log files. The log files kept records of the features visited by each subject. The data from the log files was collected weekly.
- **Screen video capture software:** Three voluntary subjects were recorded every week from weeks 2 to12. The screen recording was captured in the form of an audio/video data file (AVI format: Audio Video Interleave).
- **The DIVTIC weekly task:** The subjects were given a weekly task to complete at the beginning of a weekly session from weeks 2 to 12, except for weeks 7 and 11. This was collected by the tutors at the end of the session. There were a total of 9 weekly tasks.

All the interviews were conducted in Thai. These were first transcribed from Thai into a word document. Then, with the help from an Educational Consultant & Translator, Ms Catherine Samananda, the interviews were then translated into English. Catherine was born and raised in Thailand. She has lived in Perth, Western Australia for many years. She graduated in Business and holds postgraduate qualifications in Professional

Development in Higher Education - UNSW. She is also well qualified and experienced in managing the affairs of Interpreting and Translating under the National Accreditation Authority for Translators and Interpreters (N.A.A.T.I.). She has conducted courses in Thai language and Culture for many tertiary institutions in Western Australia. The interview translation took her approximately 30 hours to complete.

5.7 Analysis of Data

All collected data were analysed as follows.

1. Laboratory tests one and two: Individual scores for each student from both groups was entered into Excel spreadsheets for analysis with SPSS.
2. Midterm and final examinations: Individual scores for each student from both groups was entered into Excel spreadsheets for analysis with SPSS.
3. DIVTIC self-administered evaluation questionnaire form: All 33 questions of the Likert scale in part 1 and all questions in part 2 for each student in the experimental group were entered into Excel spreadsheets for analysis with SPSS.
4. Subject semi-structured interview form: Each interview was categorised to demonstrate major patterns in the descriptive analysis.
5. Tutor semi-structured interview form: Each interview was categorised to demonstrate major patterns in the descriptive analysis.
6. Tutor observation form: Each Tutor observation form was categorised to demonstrate major patterns in the descriptive analysis.
7. Researcher observation form: Each observation form was categorised to demonstrate major patterns in the descriptive analysis.
8. DIVTIC weekly task: The results of each DIVTIC weekly task for each student was recorded and coded for analysis with SPSS.
9. DIVTIC log in record: Each log-in record for each student in the experimental group was inserted into Excel spreadsheets for analysis with SPSS.
10. C WebBoard: Each WebBoard message was categorised to demonstrate major patterns in the descriptive analysis.

5.8 Reliability and Validity of Data Collection

Aiken (1997, p. 165) writes that "An instrument must be reliable in order to be valid, but it is not necessarily valid because it is reliable." To ensure that the collected data was reliable and valid, this study used the technique of triangulation as described in Chapter 5.1.3. In the social sciences, triangulation used to view of data from more than one standpoint (Burns, 1994). For example, this study compared and contrasted the data collected from the DIVTIC self-administered evaluation questionnaire form and the DIVTIC log file for the time spent in DIVTIC. Multiple questions were also used to assess attitudes such as useability, level of enjoyment and some other affective variables. Moreover, the collected data from the Tutor observation form was compared to the DIVTIC self-administered evaluation questionnaire form which was completed by the subjects and was used to support the validity and reliability. Furthermore, tutors were asked to walk through and collect the DIVTIC self-administered evaluation questionnaire form from the subjects at the end of the session to ensure that each report was answered by the individual subject.

5.9 Ethical Considerations

In the first class, the researcher explained the reasons for the laboratory sessions and his research which sought to explore how students learned introductory computer programming courses via the use of a computer-based learning aid, namely DIVTIC. The subjects were also informed of the importance of the study and if, for any reason, they felt uncomfortable with the study, they would be able to withdraw their consent at any time without penalty.

The privacy of the subjects was a concern for the researcher as Merriam (1998) states

> Interview—whether it is highly structured with predetermined questions or semistructured and open-ended—carries with it both risks and benefits to the informants. Respondents may feel their privacy has been invaded, they may be embarrassed by certain questions, and they may tell things they had never intended to reveal. (p. 214)

To protect the subjects' right of privacy, the data was treated with the strictest confidence. The subjects were not identified by name in any reports. The interviews, which were transcribed verbatim, were stored on a computer in a private and secure location. Furthermore, all data collected will be destroyed 5 years after the completion

of the study. Finally, a consent form was given to those subjects who indicted that they were willing to be part of the study.

CHAPTER 6: RESEARCH QUESTION 1

Two research questions were proposed in Chapter 3. This chapter discusses research question 1, while Chapter 7 discusses research question 2. Table 6.1 shows the question, method, data collection, and analysis conducted in answering research question 1, *how did students use DIVTIC?* The intention of this question was to explore how students in this study used the tool so that findings associated with learning outcomes could be understood in relation to students' usage patterns.

Table 6.1: Data matrix for research question 1

Research Question	Method	Data Collection	Data Analysis
1. How do students use DIVTIC?			
(1a) Which components of DIVTIC do students use and for how long?	(1a) A, G, and I	(1a, 1b) Collect students' attitudes and performances data from A (weeks 6 and 10) and G, and I (weekly)	(1a, 1b. 1d) Descriptive analysis
(1b) What strategies do students use with DIVTIC?	(1b) A, B, an G		
(1c) What factors influence students' use of DIVTIC?	(1c) A, B, C, D, E, and G	(1c, 1d) Collect data relating to the effects of using DIVTIC from A (week 6) and E (weeks 7 and 11) and B, C, D, and G (weekly)	(1c) Qualitative analysis to identify patterns in improvement towards DIVTIC
(1d) What attitudes do students generate towards DIVTIC?	(1d) A, B, C, and E		

NOTE:
A: DIVTIC self-administered evaluation questionnaires
B: Subject semi-structured interviews
C: Tutors' observations
D: Researcher's observations
E: Tutor semi-structured interviews
F: Lab test 1 and 2, Midterm and final examinations
G: Screen recording
H: DIVTIC weekly task
I: DIVTIC log file

6.1 Question 1a: Which Components of DIVTIC do Students Use and for How Long?

DIVTIC was comprised of eight components: Syllabus/Lecture notes, Computer Structure, Animated Examples, C Compiler, C WebBoard, Self-evaluation, FAQ Pool, and C References and Links. Students in this study were divided into three different levels according to GPA: low (1.11 to 1.72); average (1.78 to 2.22); and high (2.25 to 3.36). For example, the control group, Group C, included C1, C2, and C3 which referred to low, average, and high GPA respectively, and the experimental group, Group E, was comprised of E1, E2, and E3 which also referred to low, average, and high GPA respectively as shown in Table 6.2. Dividing the students by their GPA into three levels helped the inquiry to explore the different usages among each level.

Table 6.2: Number of students from both groups in three different GPA levels

Group	Low GPA 1.11 – 1.72	Average GPA 1.78 – 2.22	High GPA 2.25 – 3.36
Control	16	17	17
Experimental	16	17	17

The methods used to collect data to answer the first research question included the DIVTIC self-administered evaluation questionnaires, screen recordings, and DIVTIC log files. The use of each method is described in more detail below:

- **The DIVTIC self-administered evaluation questionnaire form:** This form was used twice, in weeks 6 and 10. It contained three parts: (1) questionnaire, (2) checklist, and (3) open-ended question. The questionnaire was comprised of 11 scales using a five-point Likert rating scale which questioned the students' perceptions of the impact of DIVTIC on their higher-order thinking, confidence, motivation, user friendliness, enjoyment, interest, level of boredom, useability, clarity, collaboration, and experience. Each scale was comprised of three questions which questioned aspects of the same element. The checklist part was comprised of three categories to explore: (a) how often, (b) what strategies, and (c) when the students used DIVTIC. The open-ended question was related to other potential problems in using DIVTIC. However, only category (a) in Part 2 of the checklist questions, *How many times have you used the following components of DIVTIC in the previous two weeks?*, was used to answer this question. The statistics software application, SPSS, was used to calculate frequency distributions.

- **Screen recordings:** Screen video capture software was used to record the activities of three students using DIVTIC for approximately 30 minutes each week starting from week 3 extending through to 12, excluding weeks 7 and 11, in which the laboratory tests 1 and 2 were given, respectively. In total, there were 24 screen recordings, including five students with low GPA, seven students with average GPA, and 12 students with high GPA. These screen recordings were used to explore which components in DIVTIC students used and for how long. At the beginning of each laboratory, three students were asked to volunteer to use the screen video capture software to record their screen while they were using DIVTIC. The weekly task was also given to the

students at the beginning of the laboratory. Therefore, students needed approximately 10 to 15 minutes to complete the weekly task and then they would log into the DIVTIC system to play the relevant animation for the weekly task. This was expected to last from 15 to 20 minutes.

- **DIVTIC log files:** The log-in files were recorded and updated onto the database on a weekly basis starting from weeks 2 through to 11. At the end of the study, all the log-in files were put together into one big file and separated, using a perl script, into several files by using the students' ID as a name for each file.

To answer the research question, *Which components of DIVTIC do students use and for how long?*, the following discussion describes students' usage patterns with DIVTIC components and students' time spent with DIVTIC, and it examines any patterns and relations which appeared to exist.

6.1.1 Level of Students' Usage of DIVTIC Components

In relation to the frequency of use of each component in the DIVTIC system from weeks 2 through to 10, the following discussion explores patterns and themes that emerged from the findings and in particular from the students' different learning abilities based on their GPA scores. As indicated earlier, DIVTIC was comprised of eight components. The features of each component are described below and followed by a table containing the frequency of use, an explanation and an analysis. A summary is provided and conclusions are drawn at the end of each section.

- **Use of syllabus/lecture notes**

The syllabus/lecture notes component included a set of course materials and relevant information as described in Section 4.1.1 It aimed to provide a self-regulated learning environment by allowing students to manage their own time and their own knowledge development. Students could get all information relating to the course from this source. It was expected that this would help students to save a lot of time gathering the information they needed and encourage them to visit other components having already logged into the DIVTIC system.

Table 6.3: Frequency use of Syllabus/Lecture Notes from Self-administered evaluation questionnaire

Topic	N	Missing	Week	Time (s) used in Previous Two Weeks				
				0	1-3	4-6	7-10	> 10
Syllabus/Lecture Notes	39	13	6	3 (10%)	19 (63.3%)	7 (23.3%)	0 (0%)	1 (3.3%)
	39	13	10	4 (15.4%)	12 (46.2%)	4 (15.4%)	4 (15.4%)	2 (7.7%)

Table 6.3 shows that in week 6, the majority of the students (63.3 percent) reported using the syllabus/lecture note component one to three times in the previous two weeks, seven students (23.3 percent) reported using it four to six times, one student (3.3 percent) reported using it more than 10 times, and three students (10 percent) reported not using it at all. However, in week 10 the majority of the students (46.2 percent) reported using this component one to three times, four students (15.4 percent) reported using it four to six times, four students (15.4 percent) reported using it 7 to 10 times, two students (7.7 percent) reported using it more than 10 times, and four students (15.4 percent) reported not using it at all.

The results show that the students were more likely to use the syllabus/lecture notes towards the end of the course than they were at the start. Some possible explanations of this may have come from the fact that students appeared to have more confidence in using DIVTIC as they progressed in the course leading them to know that some information was available on the web site for their inquiry. The results also revealed that few students did not use the syllabus/lecture notes at all. An interesting observation was that the majority of these students, who did not use this component at all, had a high GPA. It appeared that the syllabus/lecture notes were perceived to be of minimal value among students with a high GPA while they seemed to be beneficial for those students who had a low or average GPA. A possible explanation for this may have been that the students with a high GPA did not need such information from this component since they could receive this information in the classroom from their teachers. On the other hand, the students with a low or average GPA may not have been able to acquire all information provided in the classroom. Thus, these students may have needed to acquire some further information at their own pace.

The results show that a total of 18 students responded in both weeks 6 and 10, among whom were five students (27.77 percent) who used it less, eight students (44.44 percent) who used it consistently, and five students (27.77 percent) who used it more so toward

the end. The students with a low GPA used this feature more than those students with an average or high GPA.

- **Use of computer structure**

The computer structure component was designed to include the animated illustration that would give novice students an overview of the basic structure of a computer and provide students with the opportunity to familiarise themselves with the overall functioning of a computer, as described in Section 4.1.2. This feature would allow novice students to navigate through the animation of each part of the computer, its description and detail. Novice students might gain more confidence from using this feature which could lead them to use other components of DIVTIC as well.

Table 6.4: Frequency use of Computer Structure from Self-administered evaluation questionnaire

Topic	N	Missing	Week	Time (s) used in Previous Two Weeks				
				0	1-3	4-6	7-10	> 10
Computer Structure	39	8	6	4 (12.9%)	18 (58.1%)	6 (19.4%)	2 (6.5%)	1 (3.2%)
	39	13	10	4 (15.4%)	18 (69.2%)	1 (3.8%)	2 (7.7%)	1 (3.8%)

Table 6.4 shows that in week 6, there were four students (12.9 percent) who reported not using the computer structure component at all in the previous two weeks, 18 students (58.1 percent) reported using it one to three times, six students (19.4 percent) reported using it four to six times, two students (6.5 percent) reported using it 7 to 10 times, and one student (3.2 percent) reported using it more than 10 times. However, in week 10 there were four students (15.4 percent) who reported not using this component at all, 18 students (69.2 percent) reported using it one to three times, one student (3.8 percent) reported using it four to six times, two students (7.7 percent) reported using it 7 to 10 times, and one student (3.8 percent) reported using it more than 10 times.

From week 6 to week 10, there was no change in the number of the students who did use and who did not use the computer structure component. This component was designed to help students as a guide to understanding some basic concepts of computer structure. There were a total of 19 students who responded in both weeks 6 and 10. Of these students, there were nine students (47.37 percent) who used it less in week 10 than week 6, eight students (42.11 percent) used it consistently between the two tests, one student (5.26 percent) did not use it at all, and one student (5.26 percent) used it more toward the end. The results showed that this component provided more useful

information at the beginning of the course but was less important to students as the course progressed. A possible explanation was that as the students came to understand the basic concepts of computer structure, their use of this component was seen to decrease.

However, the results also revealed that there was one student who used this component more toward the end. A possible reason behind this increment could be that this student, who had a low GPA, did not know about the feature in the first weeks of the course and only discovered it as the course progressed. In addition, the student with an average GPA was the only one who did not use this component at all. A possible reason could be that this student may have come with some basic knowledge of computer structure or that he or she may have not noticed this existing component. The maximum use, more than 10 times, was by those students with an average GPA in week 6 and a high GPA in week 10. Overall, these results seemed to suggest that the students with an average or high GPA had more motivation than those with a low GPA. A possible reason could be that these students, with an average or high GPA, may have had more time to browse because they found the question problems less difficult and took less time to complete them.

- **Use of animated examples**

The animated examples component was a set of animation examples which students would interact with by clicking on the control buttons as described in Section 4.1.3. The animations would guide students through each step of program execution. A marker was used to animate each line throughout all segments of each line of the program. This component, which was planned to be the main function of the DIVTIC system, would encourage students to be active learners by enabling them to construct their own knowledge. Students would be given a weekly task at the beginning of each laboratory session and they would be asked to use this component after completing the weekly task to check their answers by watching a specific animation stated on the weekly task. This was to ensure that all students had used this component in every single week. Students would probably navigate through additional animation examples other than the required one. This would lead them to be active learners.

Table 6.5: Frequency use of Animated Examples from Self-administered evaluation questionnaire

Topic	N	Missing	Week	Time (s) used in Previous Two Weeks				
				0	1-3	4-6	7-10	> 10
Animated Examples	39	8	6	0 (0%)	13 (41.9%)	9 (29.0%)	5 (16.1%)	4 (12.9%)
	39	8	10	0 (0%)	14 (45.2%)	5 (16.1%)	6 (19.4%)	6 (19.4%)

Table 6.5 shows that in week 6, there were 13 students (41.9 percent) who reported using the animated example component one to three times in the previous two weeks, nine students (29 percent) reported using it four to six times, five students (16.1 percent) reported using it 7 to 10 times, and four students (29 percent) reported using it more than 10 times. However, in week 10 there were 14 students (45.2 percent) who reported using this component one to three times, five students (16.1 percent) reported using it four to six times, six students (19.4 percent) reported using it 7 to 10 times, and six students (19.4 percent) reported using it more than 10 times.

The results show that the number of students using this component gradually increased from weeks 6 to 10, except for those students who reported using it four to six times, in which the number of students decreased from 9 to 5. There were a total of 24 students who responded in both weeks 6 and 10 among whom eight students (33.33 percent) used it less, 10 students (41.67 percent) used it consistently, and six students (25 percent) used it more toward the end. The results also show that all students used this component. Similarly, the results corresponding from the screen recording also indicated that this component was used the most. These findings provide some evidence of the usefulness of this component. Some possible explanations for the usage patterns observed were that they were asked to use this component to check their answer as a part of a requirement for using DIVTIC, that the students enjoyed using this component to enhance their understanding, or that this component was the only component that was designed to incorporate an animation of a step-by-step visualisation of program execution.

The students who most used the animated example component, were those who used them more than seven times. These were the students with an average or high GPA in week 6. On the other hand, in week 10 the results show that the majority of the students who used this section more than seven times were those with a low GPA. This finding seemed to suggest that the students with a low GPA were likely to make slow progress

in learning and generally started using the animated example component towards the end of the course. These students may have found the latter parts of the course more difficult and needed DIVTIC to help them. Alternatively, it may have been that the students slowly began to realise the usefulness of these tools as the course progressed and made more use of these resources to support their learning toward the end of the course.

Another interesting pattern came from the screen recording usage data which showed that the majority of the students who played and watched the animation twice or more with interaction were students with a high GPA. This pattern seemed to suggest that the students with a high GPA enjoyed using DIVTIC as a learning tool by interacting and watching the animation more than those with a low or average GPA during the laboratory session. Some possible explanations for the observed patterns of usage observed were that students with a high GPA may have made fast progress in learning, as expected, so that they were able to complete the animation process and start it over again.

- **Use of C compiler**

The C compiler component is a step-by-step animation that was designed to provide information on how to use a C compiler. It aimed to help students become familiar with the C compiler. This feature would save students' time spent figuring out how to use the C compiler by providing each keyword and its associated description, together with a simple C source code along the way, as a hint to encourage each student to write their first simple program. Students would appreciate this feature as it could help them make faster progress in learning to program such as understanding how to use C compiler, running and saving the code, etc. Novice students would not have to spend time gathering information and studying it by themselves.

Table 6.6: Frequency use of C Compiler from Self-administered evaluation questionnaire

Topic	N	Missing	Week	Time (s) used in Previous Two Weeks				
				0	1-3	4-6	7-10	> 10
C Compiler	39	8	6	2 (6.5%)	20 (64.5%)	7 (22.6%)	1 (3.2%)	1 (3.2%)
	39	11	10	3 (10.7%)	15 (53.6%)	8 (28.6%)	1 (3.6%)	1 (3.6%)

Table 6.6 shows that in week 6, there were two students (6.5 percent) who reported not using the C compiler component at all in the previous two weeks, 20 students

(64.5 percent) reported using it one to three times, seven students (22.6 percent) reported using it four to six times, one student (3.2 percent) reported using it 7 to 10 times, and one student (3.2 percent) reported using it more than 10 times. However, in week 10, there were three students (10.7 percent) who reported not using this component at all, 15 students (53.6 percent) reported using it one to three times, eight students (28.6 percent) reported using it four to six times, one student (3.6 percent) reported using it 7 to 10 times, and one student (3.6 percent) reported using it more than 10 times.

The results show that there was a big drop in the number of the students who used the C compiler component from 20 (64.5 percent) in week 6 to 15 (53.6 percent) in week 10. This appeared to indicate that this component was seen to be more useful to students' learning at the beginning as expected. In week 6, there were only two students, with an average or high GPA, who used this component more than seven times. On the other hand, there was one student with a low GPA who used this component between 7 to 10 times and one student with an average GPA, different from the one in week 6, used it more than 10 times in week 10.

There were a total of 21 students who responded in both weeks 6 and 10 among whom six students (28.57 percent) used the C compiler component of DIVTIC less, 13 students (61.9 percent) used it consistently, and two students (9.53 percent) used it more toward the end. The results appeared to suggest that the students used this component consistently from the beginning toward the end of the course. They also seemed to suggest that the students were satisfied with the feature of this component as they did have an opportunity to acquire some information that helped them understand the features or options of the C compiler environment. They may have found that they could save a lot of time by navigating through this component rather than searching other sources.

- **Use of C WebBoard**

The C WebBoard component was designed for students to communicate with their peers. This feature would encourage individuals to share and change their ideas (Hsi, 1997) and lead them to discover, analyse, synthesise, and evaluate each other's ideas (Gokhale, 1995; Norman & Sphorer, 1996). Students would also be able to post

questions and receive answers via this feature. Moreover, it would enable students to participate in collaborative learning which is a significant factor of student learning. Students with learning difficulties could get some help from this feature by posting their programming problems and waiting for their peers to respond with answers. On the other hand, good students with a high ability in learning could share their experiences, giving some hints or tips on how to write programming code. This feature would act as a channel for students to exchange information, either when they were physically apart or when they were perhaps too shy to ask questions in front of others in class. This feature would lead them to learn more effectively by facilitating collaboration and the sharing of ideas and programming tactics.

Table 6.7: Frequency use of C WebBoard from Self-administered evaluation questionnaire

Topic	N	Missing	Week	Time (s) used in Previous Two Weeks				
				0	1-3	4-6	7-10	> 10
C WebBoard	39	8	6	12 (38.7%)	15 (48.4%)	4 (12.9%)	0 (0%)	0 (0%)
	39	12	10	8 (29.6%)	14 (51.9%)	5 (18.5%)	0 (0%)	0 (0%)

Table 6.7 shows the results from the Self-administered evaluation questionnaire form. These results were discrepant to the results from the log-in records as shown in Figure 6.1 in which all students (100 percent) had used the animated example component while no one used the C WebBoard at all during the entire study.

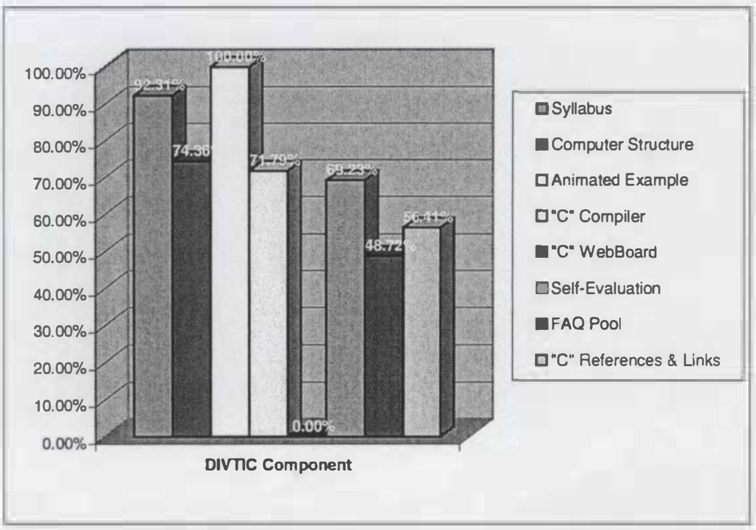


Figure 6.1: The usage of each component of DIVTIC from log files

The results revealed that the majority of the students were not using the WebBoard as a channel to communicate with their peers. Some possible explanations for the patterns of

usage observed were that students did not want to waste their time in using this component since they could discuss face-to-face with their peers during the laboratory session. Likewise, the results may indicate that they may have had no time to get involved with this component because they were busy in using other components such as the animated example component.

However, the log-in record data was more reliable as it provided a record of the times when the students visited the page. Therefore, the findings concluded that there was no activity for use of the WebBoard component at all during the entire study. It was a big surprise that students had not used this component at all. Furthermore, it was impossible to find out why the students had given false information about this in the evaluation form. The students should have noticed by week 10 that they had not used the C WebBoard at all. Some possible explanations may have come from the fact that students may have thought other students used it so they just simply gave a false answer to please their teacher. In Thai culture, it is important to respect the teacher. Perhaps students were used to the traditional teaching and learning style, behaviorism, and not familiar with the new teaching and learning style, constructivism, which encourages them to be active learners.

The results also revealed that more than 90 percent of the students used the syllabus/lecture notes and animated example components. Approximately, 70 percent of the students used the computer structure, C compiler, and self-evaluation components. The use of the C references and links component was at 56.41 percent while the FAQ Pool component was only 48.72 percent. One possible strategy to increase the use of the WebBoard would be to include a programmed topic discussion. The WebBoard probably needed more planned activities such as weekly discussion topics linked to the weekly tasks.

- **Use of self-evaluation**

The self-evaluation component was designed to allow students to test their own understanding. It was comprised of a set of multiple-choice questions which covered all topics. It provided dynamic feedback when students clicked on an answer, thus encouraging the learning process (Alam & Renci, 1998). This feature was intended to increase students' motivation to test their own understanding of each topic and to

provide them with dynamic feedback while using this component. Students would gain understanding and confidence by testing their knowledge with this feature at their own pace and in their own time.

Table 6.8: Frequency use of Self-evaluation from Self-administered evaluation questionnaire

Topic	N	Missing	Week	Time (s) used in Previous Two Weeks				
				0	1-3	4-6	7-10	> 10
Self-evaluation	39	11	6	11 (39.3%)	13 (46.4%)	2 (7.1%)	2 (7.1%)	0 (0%)
	39	14	10	8 (32.0%)	12 (48.0%)	4 (16.0%)	1 (4.0%)	0 (0%)

Table 6.8 shows that in week 6, there were 11 students (39.3 percent) who reported not using the self-evaluation component at all in the previous two weeks, 13 students (46.4 percent) reported using it one to three times, two students (7.1 percent) reported using it 7 to 10 times, and no one reported using it more than 10 times. However, in week 10, there were eight students (32.0 percent) who reported not using this component at all, 12 students (48.0 percent) reported using it one to three times, four students (16.0 percent) reported using it approximately four to six times, one student (4.0 percent) reported using it 7 to 10 times, and no one reported using it more than 10 times.

As the number of the students who used this component was stabilised from weeks 6 to 10, it seemed that students did engage in using this component. The majority of the students used this component approximately one to three times. There were only four students, one with a high, two with an average, and one with a low GPA, who used this component between 4 to 10 times in week 6, in which the students with a low or average GPA used this component the most. There was also an indication of a repeated pattern in using this component from one student with a low GPA who used it for approximately the same amount of time throughout the course. The rest of the students, one with a high and two with average GPA, did not respond in week 10. This fact seems to suggest that students were more likely to test their abilities once they had learned more and gained more or sufficient knowledge.

A total of 16 students responded in both weeks 6 and 10 among whom, three (18.75 percent) used this component less in week 10 than in week 6, six (37.5 percent) used it consistently between the two tests, four (25 percent) did not use it at all, and three (18.75 percent) used it more toward the end. The four students who did not use this component at all included three students with a low GPA and one student with an

average GPA. This finding appeared to suggest that students with a low GPA were less likely to use this component to test their understanding. A possible explanation may have come from the fact that these students appeared to have a difficult time in understanding the programming concepts. They may not have wanted to waste valuable time by doing tests which they knew would indicate their lack of knowledge. Likewise, this finding may indicate that these students may not have known about this feature. Perhaps the students' focus was based on trying to understand the concepts rather than testing their full understanding.

- **Use of FAQ pool**

The FAQ pool component contained frequently asked questions (FAQs). This feature was designed to provide students with answers to common questions that other peers have asked. This aimed to be a first point of reference for students when they had a question. Students would be able to access this feature at their own pace and in their own time. It was expected that most of the time, students would get the answer they needed to make progress in their learning.

Table 6.9: Frequency use of FAQ Pool from Self-administered evaluation questionnaire

Topic	N	Missing	Week	Time (s) used in Previous Two Weeks				
				0	1-3	4-6	7-10	> 10
FAQ Pool	39	10	6	16 (55.2%)	12 (41.4%)	1 (3.4%)	0 (0%)	0 (0%)
	39	14	10	12 (48.0%)	9 (36.0%)	4 (16.0%)	0 (0%)	0 (0%)

Table 6.9 shows that in week 6, there were 16 students (55.2 percent) who reported not using the FAQ Pool component at all in the previous two weeks, 12 students (41.4 percent) reported using it one to three times, one student (3.4 percent) reported using it four to six times, and no one reported using it more than six times. However, in week 10 there were 12 students (48.0 percent) who reported not using this component at all, nine students (36.0 percent) reported using it one to three times, four students (16.0 percent) reported using it four to six times, and no one reported using it more than six times.

As the level of use of this component was very low, between one to three times in week 6, the students used this feature less than expected. Some possible explanations may come from the fact that students may not have had any chance to navigate through this component because of the limited time allowed in the laboratory session, or that they did not notice this feature. The results also showed that there was only one student with

a high GPA who used this component between four to six times in week 6. The majority of the students, who used this component between four to six times in week 10, were those with an average or high GPA.

There were a total of 17 students who responded in both weeks 6 and 10 among whom, two (11.76 percent) used the FAQ Pool less in week 10 than in week 6, three (17.65 percent) used it consistently between the two tests, seven (41.18 percent) did not use it at all, and five (29.41 percent) used it more toward the end of the course. The results showed that this component held the highest number of students who did not use this feature at all. The majority of the students, who used it the most, were those with a high GPA. This finding seemed to suggest that either students with a low GPA did not pay enough attention to this feature or that they did not know it existed.

- **Use of C references & links**

The C references & links component was designed to assist students in advancing their knowledge by searching for relevant information on the World Wide Web. This feature provided useful URL links from a history of C programming language to an advanced C source code. The different element in the C information component were considered be useful for students with different learning abilities because they could go to any link to suit their needs. This feature was intended to encourage students to become active learners by providing them with an opportunity to search for anything of particular interest.

Table 6.10: Frequency use of C References & links from Self-administered evaluation questionnaire

Topic	N	Missing	Week	Time (s) used in Previous Two Weeks				
				0	1-3	4-6	7-10	> 10
C References & Links	39	10	6	6 (20.0%)	18 (60.0%)	4 (13.3%)	2 (6.7%)	0 (0%)
	39	14	10	14 (53.8%)	8 (30.8%)	4 (15.4%)	0 (0%)	0 (0%)

Table 6.10 shows that in week 6, there were six students (20.0 percent) who reported not using used the C references and links component at all in the previous two weeks, 18 students (60.0 percent) reported using it one to three times, four students (13.3 percent) reported using it four to six times, two students (6.7 percent) reported using it 7 to 10 times, and no one reported using it more than 10 times. However, in week 10, there were 14 students (53.8 percent) who reported not using this component at all, eight

students (30.8 percent) reported using it one to three times, four students (15.4 percent) reported using it four to six times, and no one reported using it more than six times.

As the number of students who used this component decreased from weeks 6 to 10, this result appeared to suggest that students may not have relied upon consulting this external help mechanism as time progressed. One student with a low GPA used the references and links component from 7 to 10 times in week 6 and this student still used it a few times towards the end. These usage patterns seemed to suggest that this component was useful to the students throughout initial weeks but less so as the course progressed.

A total of 19 students responded in both weeks 6 and 10 among whom 10 (52.63 percent) used the references and links component less in week 10 than in week 6, four students (21.05 percent) used it consistently between the two tests, four students (21.05 percent) did not use it at all, and one student (5.26 percent) used it more toward the end. The students, who consistently used this component, were those with an average or high GPA. There was only one student with an average GPA, who used it more toward the end of the course but did not use this component in the first weeks. The results showed that students did not use this component toward the end of the course. Possible explanations may be that students already had such a difficult time in learning programming that they did not have time to search for any more information outside the classroom or, perhaps students had a difficult time finding useful information to solve their particular problems thus decreasing their use of the reference and links section.

- **Conclusion to level of students' usage of DIVTIC components**

The students were more likely to use all DIVTIC components except the C WebBoard as results from the log-in records showed that no one had visited the C WebBoard component at all during the entire study. Figure 6.2 (same as Figure 6.1) shows the percentage of total use for each component.

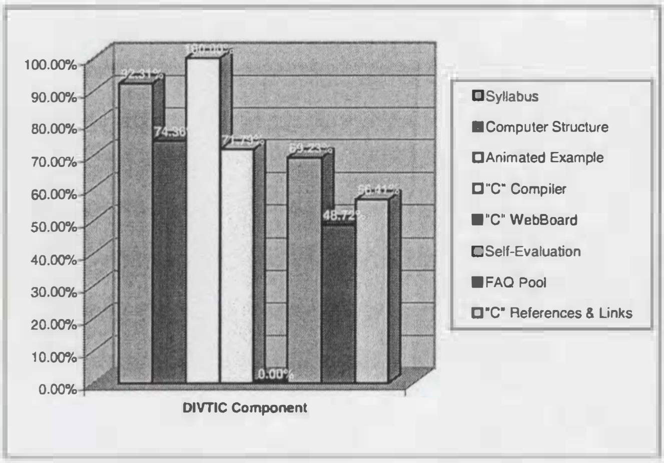


Figure 6.2: Total use for each component of DIVTIC from log files

Every component was used: the C WebBoard had the least use while the animated examples had the most use. The use of animated examples was highest, as expected, because students were mandated to use this component in every single laboratory session after they finished the weekly task. On the other hand, the FAQ Pool was not obligatory as students had the opportunity to navigate through any component they wanted in their own time and leisure. Perhaps students did not use the C WebBoard to post their enquires as they already had time to discuss problems face-to-face with their peers in the laboratory session. Many students never visited this component. They seemed too shy away from posting their enquires on the web or even using information from this component.

6.1.2 Level of Students' Time Spent with DIVTIC Components

All usage of DIVTIC was recorded into log files. The following figure (Figure 6.3) shows the time each user spent using DIVTIC from weeks 2 to 11. The minimum log-in time was 3 minutes, the maximum was 1304 minutes, and the average was 357 minutes.

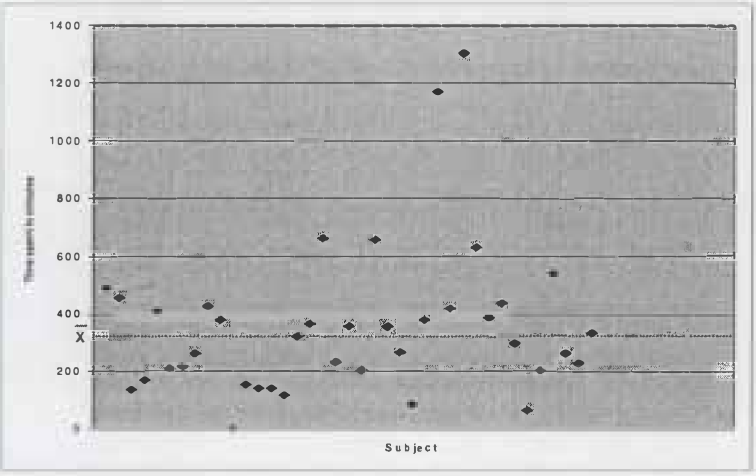


Figure 6.3: Time spent using DIVTIC for each user

The longest log-in time was for 21 hours and 44 minutes by a student with an average GPA. This student normally logged into the DIVTIC system outside the laboratory session. However, there was no evidence to show that this student had forgotten to log out. Each log in showed this student's movement from one page to another. The second longest log-in time lasted for 11 hours and 2 minutes by a student with a high GPA. This was a big jump from the first longest one. A possible explanation for this matter was that the first longest log-in time was done gradually and consistently outside and inside the laboratory session and the second longest log-in time was done by the student with a high GPA who may have taken less time to understand the material from the DIVTIC system. The following figure (Figure 6.4) shows the total log-in time of all students in each week from weeks 2 to 11.

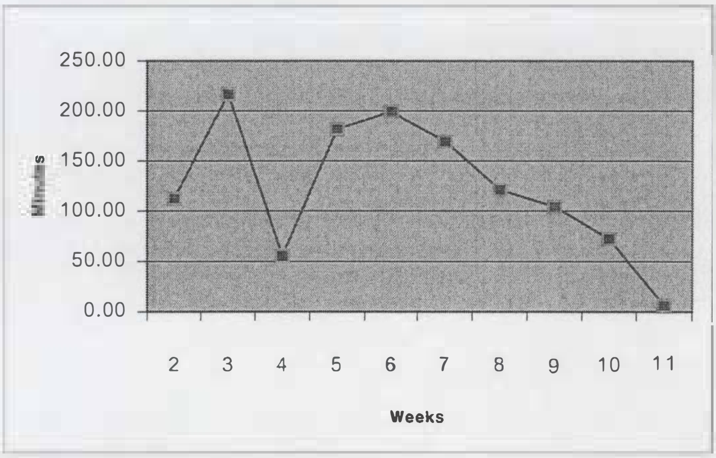


Figure 6.4: Time spent using DIVTIC each week

Figure 6.4 shows that time spent using DIVTIC in week 4 had obviously dropped from week 3. This was unexpected. The topic in week 4 was all about control statements, which was a difficult topic and more time consuming than previous topics. This seemed to be the reason why the log-in time in this week obviously went down. However, the log-in times in weeks 5 and 6 had increased sunstantially from week 4 although the materials were getting more and more complex in concept. Students may have become familiar with DIVTIC by now and wanted to use it to help them in their learning process. The log-in times in weeks 7 to 11, rapidly decreased as the course progressed. This was expected as the students had most trouble overcoming difficulties in their learning progress, weekly task, and weekly problem. These findings seem to suggest that students would use DIVITC less when they came to a stage where they could not comprehend a difficult topic.

The following figure (Figure 6.5) shows an overview of the log-in time for each group of students based on their learning abilities.

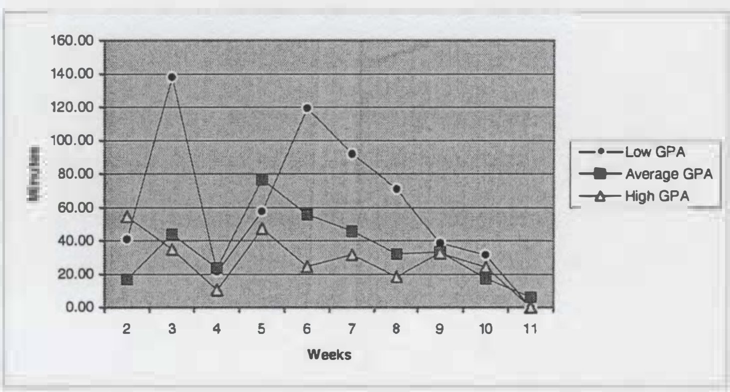


Figure 6.5: The weekly time spent using DIVTIC of students with a low, average, or high GPA

Figure 6.5 shows that the longest average log-in time during the first weeks was from the students with a low GPA. This was an expectation from the study which specifically aimed at helping students with a low ability to learn. The students with a low GPA seemed to use DIVTIC more than others, gradually increasing during the entire study, except in week 1 where the students with a high GPA used it a little bit longer and in week 4 where the students with an average GPA had used it more. These results seemed to suggest that students with low GPA needed more help with their learning.

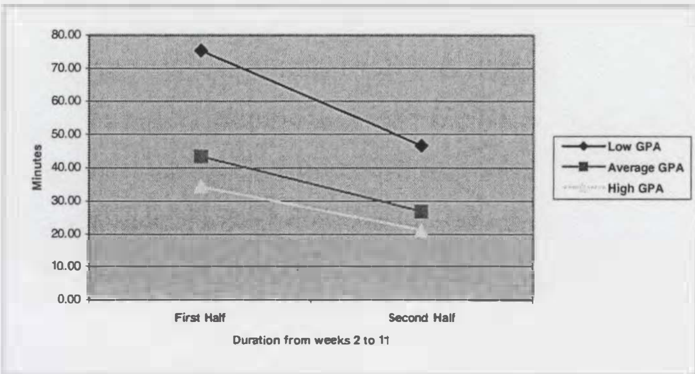


Figure 6.6: Weekly average time spent using DIVTIC between first and second halves

Figure 6.6 show that the highest frequency of log-ins occurred during the first half of the study from weeks 2 to 6. The majority of these students had a low GPA. The results seemed to suggest that students with a low GPA found DIVTIC more useful in helping them understand the concepts and in preparing for the test. The log-in time was less in the second half of the study from weeks 7 to 11 as all students used DIVTIC less as the course progressed. A possible explanation for the patterns of usage observed were that all students may have encountered learning difficulties as the course progressed and the learning materials became more complex.

The first time the students' logged in, there were 12 students (30.77 percent) who logged into the syllabus/lecture notes component. There were nine students (23.05 percent) who went to the computer structure component and eight students (20.51 percent) who went to the animated examples component. The rest of the students (25.67 percent) went to other components including the FAQ Pool, the C compiler, and the self-evaluation. This pattern seemed to suggest that the information such as a course outline, lecture notes, samples, etc. in the syllabus/lecture notes component was all necessary information that should not be excluded from the web site. The basic computer structure also appeared to be an important function for the students as it indicated to be a second most-often-used component in the first log-in time. The results suggested that the students appeared to gain some benefits from providing this information because they could access at their own time and pace. The students appeared to be satisfied with these provided resources.

The students used all the components of DIVTIC except for the WebBoard. However, the results show that there were only three major components which the students used

frequently throughout the study. These were the animated example, the syllabus/lecture notes, and the self-evaluation components. The students spent most of the time in using the animated example component while there was no student who spent time in using the WebBoard. The second most-often-used component was the syllabus/lecture notes component and the third one was the self-evaluation component.

The results show that the syllabus/lecture notes component appeared to be useful and more important towards the end of the course as students progressed and came to know that this section provided some relevant information for their inquiry. On the other hand, the results also showed an opposite pattern for the use of the computer structure and the C compiler components of DIVTIC. Students appeared to use these components less as the course progressed. The students with an average or high GPA appeared to use this component more than those with a low GPA. These components, however, appeared to be useful only for the first weeks of the course. Once the students understood basic computer structure and the features available in C compiler, they tended to use it less.

The average frequency of the use for the animated examples component was the highest one and used up to 252 minutes, the syllabus/lecture notes component was 50 minutes, and the self-evaluation component was 39 minutes. The average use for the C compiler component was 6 minutes, the computer structure component was 4 minutes, the FAQ Pool component was 3 minutes, and the C references and links component was 2 minutes as shown in Figure 6.6. The high levels of use of the animated examples component was expected as it was designed as a major feature of the DIVTIC system.

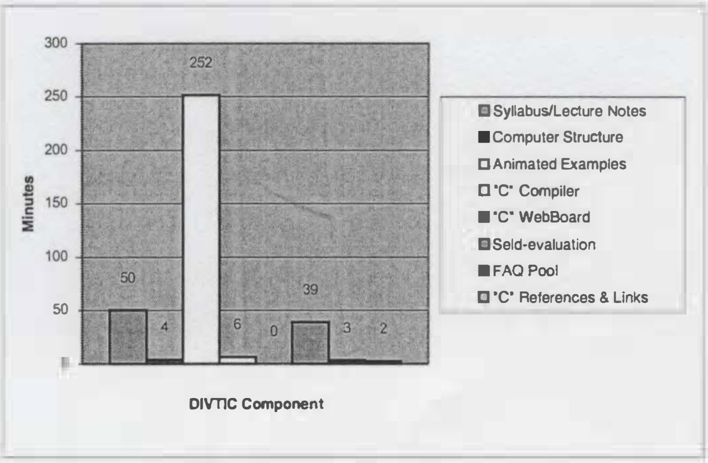


Figure 6.7: The average time use for each component of DIVTIC from log files

The following table (Table 6.11) shows the components students used and time spent using each component of DIVTIC as demonstrated by data collected from the screen recordings.

Table 6.11: Component use and time spent from screen recording

Component in DIVTIC				Other	
Animated Example		Syllabus/ Lecture Notes			
Student	Average (minute)	Student	Average (minute)	Student	Average (minute)
24 (100%)	26.40	2 (8.33%)	2.70	8 (33.33%)	4.02

Data taken from the screen recordings revealed that there were only two components, the animated example and syllabus/lecture notes components, being used when this data was collected (Table 6.11). Within the screen recordings, there were 24 students (100 percent) who used DIVTIC with an average of 26.40 minutes, which was over the expected average of between 15 to 20 minutes. There were only two students (8.33 percent) who used the syllabus/lecture notes component with an average of 2.70 minutes. There were also eight students (33.33 percent) who used other programs or links including C IDE (Integrated Development Environment, three students), Yahoo web site (one student), Karaoke web site (one student), and Thai Dictionary program (three students) while they were using DIVTIC. The average of the use of other features was 4.02 minutes.

The results appeared to verify that during the laboratory session, the animated example component was used the most. This was as expected since the students were asked to run the relevant animation inside the animated example component to check their answers. However, some unexpected results were also discovered, for example, there were also two students who searched and did something else beside the relevant learning activities. One of them who had an average GPA went to the Yahoo web site to read and send e-mail while another one who had a low GPA went to a Karaoke web site.

Two students who both had a low GPA used the syllabus/lecture notes component. The students with a low GPA seemed to need more information on course materials than those with an average or high GPA. The students with an average and high GPA used a Thai dictionary program alongside the use of DIVTIC. This seemed to indicate that these students had more motivation in acquiring new knowledge through the use of other media or software. Three students who chose to see the output without watching

the entire animation were at all different levels: low, average, and high GPA. These recordings were undertaken for the first time in week 3. However, the following weeks showed that all students used DIVTIC by watching the entire animation with and without interacting with the animations. Nobody jumped immediately to the end to see the output as happened in week 3, the first week of recording. A possible reason for this may have been that the students felt more likely to use DIVTIC by either watching or interacting with the animation process. They may have felt that the animation process helped them to learn how to program in a more effective way.

On the other hand, by watching the students use DIVTIC from the screen recordings, it appeared that students with a low GPA did not pay enough attention to DIVTIC. In the first week of recording in week 3, one student went to use an e-mail program rather than using DIVTIC. These students with a low GPA seemed to learn only from whatever information or materials were provided by the teacher without gathering any further knowledge from other sources. However, interactions increased towards the end of the course when students tended to play and interact more with DIVTIC. The students with a high GPA appeared to have more interactions than those with an average or low GPA. In addition, there were 11 students who navigated through other animations beside the suggested ones. These students had a majority with an average or high GPA. This result appeared to suggest that students with a low GPA seemed to have less motivation than those students with an average or high GPA or that they may have already faced learning difficulties and had no time to navigate to other animations beside the suggested ones.

- **Conclusion to students' time spent with DIVTIC**

The longest log-in time lasted for 21 hours and 44 minutes by a student with an average GPA. This student normally logged into DIVTIC outside the laboratory session. Each log in of this student showed movement in activities from one page to another. The second longest log-in time lasted for 11 hours and 2 minutes by a student with a high GPA. While these results are inconclusive, the results coupled with other data seemed to suggest that students with a low GPA found DIVTIC useful in helping them understand more about the concepts and as well as helping them to prepare for the test.

The animated examples component was the highest used with up to 252 minutes of use; the syllabus/lecture notes component was 50 minutes; and the self-evaluation component was 39 minutes. The average for using the C compiler component was 6 minutes; the computer structure component was 4 minutes; the FAQ Pool component was 3 minutes; and the C references and links component was 2 minutes. From the screen recording, there were only two components, the animated example and the syllabus/lecture notes components, in the DIVTIC system that were used. A total of 24 students (100 percent) used the animated examples component with an average of 26.40 minutes, which was over the expected average of between 15 to 20 minutes, while two students used the syllabus/lecture notes component with an average of 2.70 minutes.

6.1.3 Conclusions

The animated examples component appeared to be the most often-used component in which the students with a low GPA made more use of this feature to support their learning toward the end of the course. However, there was an opposite pattern in the use of the WebBoard component to the animated example component. The results from the log files show that no one had gone to the WebBoard page throughout during the entire study. It seemed to suggest that the use of the WebBoard as part of a constructivist learning environment was not appropriate with Thai students who were still familiar with a more traditional behaviorist teaching and learning style. In addition, the results appeared to conclude that students with an average or high GPA seemed more likely to test their understanding via the use of the self-assessment component of DIVTIC than those with a low GPA. The self-assessment component seemed to be useful for students to test their understanding as the course progressed since the average time spent using this component was obviously high at 39 minutes. However, the FAQ pool and the C reference and links components were perceived to be of minimal value among students since they had not frequently used these components and even some students did not even bother to use it at all. The students appeared to make most use from components that their teachers encouraged them to use, for example, the animated examples component.

6.2 Question 1b: What Strategies do Students Use with DIVTIC?

As indicated earlier in Section 6.1, the DIVTIC self-administered evaluation questionnaire form contained 3 parts: (1) questionnaire, (2) checklist, and (3) open-ended question. The checklist part was comprised of three categories to explore: (a) how often, (b) what strategies, and (c) when the students used DIVTIC. Only 2 categories, (b) and (c) in part 2, checklist, would be used to explore this research question. The statistics software application, SPSS, was used to calculate frequency distributions. Another method used was the Subject semi-structured interview. The use of each method is described as follows:

- **The DIVTIC self-administered evaluation questionnaire form:** This form was used twice, in weeks 6 and 10. Category (b) *How do you use Animated Example?* in the checklist part was used to explore what strategy students use the Animated Example component by composing of seven different checklist options as follows:
  - ❑ I just watch the animation without interacting with it.
  - ❑ I press ***Stop/Play*** button to think about and change what is going on.
  - ❑ I repeat the animation to make it clear how the program executes.
  - ❑ I press ***Backward/Forward*** to see the animation.
  - ❑ I stop the animation and discuss with my peers.
  - ❑ I go to test my own code right after watching the animation to compare the output.
  - ❑ Other:..

  Category (c) *When do you normally use DIVTIC?* in the checklist part was also used to explore when students use the Animated Example component comprised of seven different checklist option as follows:
  - ❑ When I am in a laboratory session.
  - ❑ When I have free time.
  - ❑ When I do assignments.
  - ❑ Before examinations.
  - ❑ When I face with programming problems.
  - ❑ When I am with friends.
  - ❑ Other:..

- **Subject semi-structure interview:** There were 24 interviews in total throughout the study. Each interview took approximately 5 minutes. A semi-structured interview format was used, which was comprised of five questions to explore how the Animated Example component help students to understand algorithms as follows:
  - Feedback on how many problems students watched?
  - Did students finish all problems in the weekly task?
  - Ask some questions about problems itself, e.g., ask about the algorithm to see if students actually learned?
  - Which part of the animation helped students learn the most and why?
  - What should be improved to make DIVTIC a better tool?

  The summary of all 24 interviews was categorised into six patterns: (a) Finished all problems, (b) Useful/Interesting, (c) Difficulty in language, (d) Most useful part, (e) Understanding the problems, and (f) Suggestions. However, only pattern (d), the most useful part pattern, was used for exploring this question.

The strategic use of DIVTIC could be divided into 3 themes, (a) Place and time usage patterns, (b) Students' interaction with DIVTIC, and (c) Animation component usage patterns, as discussed in the following sections.

6.2.1 Place and Time Usage Patterns

Table 6.12: DIVTIC usage data (Weeks 6 and 10)

Place and Time	Week 6 (N = 39, Missing = 8)		Week 10 (N = 39, Missing = 8)	
	No	Yes	No	Yes
1. When I am in a laboratory session.	6 (19.4%)	25 (80.6%)	3 (9.7%)	28 (90.3%)
2. When I have free time.	17 (54.8%)	14 (45.2%)	14 (45.2%)	17 (54.8%)
3. When I do assignments.	24 (77.4%)	7 (22.6%)	16 (51.6%)	15 (48.4%)
4. Before examinations.	18 (58.1%)	13 (41.9%)	18 (58.1%)	13 (41.9%)
5. When I face with programming problems.	14 (45.2%)	17 (54.8%)	17 (54.8%)	14 (45.2%)
6. When I am with friends.	28 (90.3%)	3 (9.7%)	30 (96.8%)	1 (3.2%)
7. Other	31 (100%)	0 (0%)	31 (100%)	0 (0%)

In week 6, Table 6.12 shows that there were 25 students (80.6 percent) who reported using DIVTIC while they were in the laboratory session whereas six students (19.4 percent) reported not using it. It also shows that 17 students (54.8 percent) reported not using it when they had free time while 14 students (45.2 percent) did. The majority of the students (77.4 percent) reported not using it when they did the assignment or (58.1 percent) before examination. There were 17 students (54.8 percent) who reported using it when they were faced with a programming problem. The majority of the students (87.1 percent) reported not using it when they were with friends or had nothing to do.

In week 10 the number of the students who reported using DIVTIC during the laboratory session increased from 25 to 28 (80.6 to 90.3 percent) whereas the number of students who reported using it when they had free time, interchanged between week 6 (no = 17, yes = 14) and week 10 (no = 14, yes = 17). It revealed that as the course progressed and the students were familiar with DIVTIC, they seemed likely to use DIVTIC more and more toward the end of the course. There was a significant increase in the number of students (7 to 15 or 22.6 to 48.4 percent) who reported using it when they did the assignments. However, there was no change in the number of the students who reported using it before the examination. The number of the students, who reported using DIVTIC when faced with programming problems, was also interchanged between week 6 (no = 14, yes = 17) and week 10 (no = 17, yes = 14). The result seemed to suggest that as the materials became more and more complex in concepts, the students were more likely to not want to use DIVTIC. A possible explanation may have come from the fact that DIVTIC was designed as a tool to help students learn how to program by visualising and understanding the programming process rather than providing results to solve their specific problems.

In relation to what situation the students' use of DIVTIC from weeks 6 to 10, the results show that there was a total of 24 students who responded in both weeks 6 and 10, including five students (20.83 percent) with a low GPA, 10 students (41.67 percent) with an average GPA, and nine students (37.5 percent) with a high GPA. The following discussion is provided to explore patterns and themes that emerged from the findings and in particular from the students' different abilities in learning based on their GPA scores. A summary and some conclusions are drawn from this discussion.

- **Using in the laboratory**

All students with a low GPA reported using DIVTIC in the laboratory in week 6, but one of them reported not using it in week 10. There were 8 out of 10 students with an average GPA who reported using DIVTIC in weeks 6 and 10. One student reported not using it at all. There were 9 out of 10 students with a high GPA who reported using DIVTIC in week 6 and all of them reported using it in week 10. The results appeared to suggest that students with a low GPA enjoyed using DIVTIC during the laboratory session in the first few weeks while the students with a high GPA had more motivation and enjoyment as the course progressed than those students with a low or average GPA by making more use of it toward the end of the course. However, the students with an average GPA seemed to use DIVTIC consistently between the two tests.

This finding may be explained in that students with a low GPA may have found the usefulness of DIVTIC to be useful when the materials had less complex concepts such as towards the end of the course. Students with an average GPA appeared to consistently use DIVTIC throughout the entire course. On the other hand, students with a high GPA may have found that the use of DIVTIC had challenged them in learning and solving problems with more complicated tasks, as was expected.

- **Using when having free time**

The results show that there were 7 out of nine students with a high GPA who reported not using DIVTIC when they had free time in both weeks 6 and 10. There were 4 out of 10 students with an average GPA who also reported not using it when they had free time in week 6 and three students reported not using it in week 10. Two out of five students with a low GPA reported not using it when they had free time in both weeks 6 and 10. Overall, about 50 percent of the students reported not using DIVTIC when they had free time. However, the pattern of usage also show that most of the students, who normally used DIVTIC when they had free time in week 6, were the same students who continually used DIVTIC in week 10.

Some possible explanations for the patterns of usage observed were that students may have been faced with difficulty in accessing the Internet to log into the DIVTIC web site in their own time and at their own pace. Students seemed to have no time outside the

laboratory session to use DIVTIC since they were required to register for 17 credits in this particular trimester in order to graduate in four years as expected.

- **Using when doing assignments**

The results show that one student with a low GPA, three students with an average GPA, and two students with a high GPA, reported using DIVTIC when they did the assignment in week 6. The numbers of students with a low, average, and high GPA who reported using DIVTIC in week 10 were increased from 1 to 2, 3 to 5, and 2 to 4, respectively. There were 12 out of 24 students who reported not using DIVTIC while they did the assignment during the entire course.

The findings suggest possible reasons why students, who reported using DIVTIC when they did the assignment throughout the entire course, may have found the use of DIVTIC was useful. It helped them to solve their assignment problems in the first weeks so that they kept using it as a tool to assist their learning progress. The number of students increased as the course progressed. On the other hand, students who reported not using DIVTIC at all while they did their assignments had no chance to find out that the use of DIVTIC may have helped solve problems with their assignments.

- **Using prior to examinations**

The results show that three students with a low GPA, three students with an average GPA, and four students with a high GPA, reported using DIVTIC before they had the examination in week 6. The numbers of students with a low or high GPA who reported using DIVTIC before the examination in week 10 decreased from 3 to 2 and 4 to 2, respectively while the number of students with an average GPA increased from 3 to 4. A total of four students reported using DIVTIC before the examination in both weeks 6 and 10. Among them there was one with a low GPA, two with an average GPA, and one with a high GPA.

Some possible explanations for the patterns of usage observed were that students with a low GPA may have found that the use of DIVTIC helped them to understand more of the basic concepts but not the complicated ones so that they used it less as the course progressed. However, students with an average or high GPA who reported using DIVTIC before the examinations may have understood some of the basic programming concepts so that the majority did not need to use DIVTIC before the examination, as

was expected. The design of DIVTIC was based on general basic programming concepts that may have not been suitable for students with an average or high GPA.

- **Using when facing programming problems**

The results show that two students with a low GPA, seven students with an average GPA, and four students with a high GPA, reported using DIVTIC when they were faced with programming problems in week 6. The number of students with a low GPA who reported using DIVTIC when they were faced with programming problems was consistent as the course progressed. However, the numbers decreased in both students with an average or high GPA from 7 to 4 and 4 to 2, respectively. Four students with an average GPA, who reported using DIVTIC in week 10 were the same students who reported using it in week 6. Likewise, two students with a high GPA, who reported using DIVTIC in week 10 were also the same students who reported using it in week 6.

These findings seem to suggest possible reasons why all students may have found that the use of DIVTIC helped them to understand and solve some basic programming problems, but not the complicated ones. However, students with a low GPA may have thought that they could use DIVTIC to help them solve complicated programming problems while other students with an average or high GPA may have thought otherwise. Students with an average or high GPA may have been successful in solving basic and complicated programming problems as a consequence of consistent use of the same students between weeks 6 and 10. This was an expectation since the design of DIVTIC was aimed at helping students with low learning abilities by providing some basic programming problems.

- **Using when being with friends**

The results show there were only three students with an average GPA who reported using DIVTIC when they were with friends in week 6 while no one reported using it in week 10. The students with a low or high GPA reported not using DIVTIC at all during the entire course when they were with friends. The findings seemed to suggest possible explanations such as that the majority of students may have preferred to work individually rather than having partners, or that the use of DIVTIC may have become to be of minimal value when they were with friends as they could do anything else. This

was to be expected since the traditional teaching and learning style, based on behaviorism, was still being used with these students.

- **Conclusion to place and time usage patterns**

In conclusion, the patterns appeared to suggest that DIVTIC was more likely to engage students when they were in the laboratory session and when these were required to use DIVTIC alongside the weekly tasks. About 50 percent of the students seemed to allocate no time to DIVTIC when they had free time. Students with an average GPA appeared to use DIVTIC most when they had free time, faced with programming problems, when doing the assignment, and when staying with friends while students with a low GPA appeared to use DIVTIC the most before the examination. The students with a high GPA seemed most likely to use DIVTIC for the first weeks of the course. However, the majority of the students, especially the students with a low or high GPA, appeared not to use DIVTIC when they were with friends. They did tend to prefer working individually. DIVTIC seemed to be a valuable resource for students who were faced with problems in basic programming concepts.

6.2.2 Students' Interaction with DIVTIC

Table 6.13: Strategy for using DIVTIC from category (b)

Strategy	Week 6 (N = 39, Missing = 8)		Week 10 (N = 39, Missing = 8)	
	No	Yes	No	Yes
1. I just watch the animation without interacting with it	22 (71.0%)	9 (29.0%)	28 (90.3%)	3 (9.7%)
2. I press **Stop/Play** button to think about and change what is going on.	9 (29.0%)	22 (71.0%)	5 (16.1%)	26 (83.9%)
3. I repeat the animation to make it clear how the program executes.	10 (32.3%)	21 (67.7%)	5 (16.1%)	26 (83.9%)
4. I press **Backward/Forward** to see the animation.	9 (29.9%)	22 (71.0%)	12 (38.7%)	19 (61.3%)
5. I stop the animation and discuss with my peers.	15 (48.4%)	16 (51.6%)	14 (45.2%)	17 (54.8%)
6. I go to test my own code right after watching the animation to compare the output.	27 (87.1%)	4 (12.9%)	31 (100%)	0 (0%)
7. Other	29 (93.5%)	2 (6.5%)	31 (100%)	0 (0%)

Table 6.13 shows that in week 6, there were 22 students (71 percent) who interacted with DIVTIC by pressing the *Stop, Play, Backward,* or/and *Forward* buttons to control the animation. Twenty-one out of 31 students (67.7 percent) had repeated the animation to make it clear and to understand each step of the program execution. There was one more student in the group who stopped to discuss with friends (16 students) than those

who did not stop the animation (15 students). The table also shows that 27 students (87.1 percent) did not bother to test their own C code to compare the output. There were only two students who did so. One described that they wanted the explanation message to be an audio message and another one stated, *"I want to take some problems programs in Animated Examples because it show very well and understand so good."*

On the other hand, in week 10, Table 6.13 shows that there were 28 students (90.3 percent) who had interacted with the animation while they were watching, but only three students (9.7 percent) who did not. There were 26 students (83.9 percent) who played and stopped the animation to make sure they understood, but only five students (16.1 percent) who did not. There were 19 students (61.3 percent) who pressed the *Stop* and *Play* buttons to think about the animation process while 12 students (38.7 percent) did not. There were 17 students (54.8 percent) who discussed the animation with their peers. Finally, all of the students (100 percent) did not test their own code right after watching the animation.

The results appeared to produce fewer interactions toward the end of the study but more thought about the animation process by responding with a bigger number when they were asked *"I press Stop/Play buttons to think about and change what is going on."* The results also show that the students felt more likely to play the animation over and over again to make it clear to themselves but ignored to press the Backward/Forward buttons to review and reflect on their understanding. The number of students who stopped the animation and discussed with their peers was consistent from weeks 6 to 10. Moreover, the time for using DIVTIC in the laboratory appeared to be inadequate. They did not appear to have time to test their own C source code right after watching the animation.

In relation to the students' use of the animated examples component from weeks 6 to 10, the results show that a total of 24 students from the screen recordings responded in both weeks 6 and 10 including five students (20.83 percent) with a low GPA, seven students (29.71 percent) with an average GPA, and 12 students (50 percent) with a high GPA. The following discussion is provided to explore the patterns and themes that emerged from the findings and in particular from the students different abilities to learn based on their GPA scores. The low interaction is used to represent an action when students used with the animation example component by simply clicking Play and Stop

buttons to control the animation process or repeated the animation more than once. A summary and further conclusions are drawn from this discussion.

- **Using animated examples with low interaction**

The results showed that three students (60 percent of students with a low GPA), two students (28.57 percent of students with an average GPA), and three students (25 percent of students with a high GPA) used the animated example component with little interaction. The findings suggest that the possible reasons for the patterns of usage observed were that students with a low GPA may have learning difficulties so that all of them appeared to have less interaction with the animation process. Students with an average or high GPA, who seemed to make fast learning progress, may have found that they needed to interact with the animation process more to explore the more complex concepts as the course progressed.

- **Using animated examples with high interaction**

The results show that many students with an average or high GPA used the control buttons (e.g., Stop, Play, Backward, and Forward) the most to interact with the animation process throughout the entire course while the students with a low GPA used it less. This may explain why students with a low GPA may have made slow learning progress in that they just simply watched the animation process without having more chance to use the control buttons to reflect on their thoughts along the way. Students with an average or high GPA may have had a better understanding of the programming concepts than those with a low GPA. As the course progressed, the complexity of the programming concepts may have proved challenging to the students with an average or high GPA and so they tended to use the control buttons more toward the end of the course.

- **Repeating the animation process**

The results show that students with a high GPA appeared to repeat the animation more than the students with a low or average GPA. There was one student with a low GPA and one student with an average GPA who did not repeat the animation at all for the entire course. A possible reason was that students with a high GPA may have made faster progress than those students with a low or average GPA, so that they may have had more time to repeat the animation process.

- **Stopping the animation process for discussion with peers**

The results showed that students with an average GPA appeared to stop the animation process to discuss with their peers more in the first weeks than toward the end of the course. The number of these students was significantly higher than those with a low or high GPA in both weeks 6 and 10. There were about 40 percent of students with a low or high GPA who appeared to consistently stop the animation process as the course progressed while the rest of them had not stopped it.

From the researcher's observation, the findings suggest that students with a low GPA may have had some difficulties in understanding the animation process and they may have felt that they were too shy to ask or discuss with their peers. On the other hand, the students with an average GPA may have had more confidence when they needed to ask a question or discuss something with their peers. Such students may have known their abilities and felt comfortable in discussing programming with their peers.

Some students with a high GPA may have thought that they were strong and fast learners so that they were less inclined to discuss things with other students who were either of the same level of ability or less. Some students with a high GPA may have felt that they would have learnt more if they had a chance to discuss problems with their peers to discover what could be useful from each situation.

- **Testing own code after watching the animation**

The results revealed four students who claimed to test their own C code right after watching the animation in week 6, among whom were two students with a low GPA, one student with an average GPA, and one student with a high GPA. As the course progressed, the results also showed that no one at all tested the code. One possible reason could be that students did not have their own source code to test at that particular time or that they did not have any source code that was similar to the animation to which to test and compare. By testing their code right after watching the animation, students would be able to compare any segment of their code to DIVTIC if they had problems with the code.

Reasons for this may have been that students may have been excited to use the animation tool in the first weeks of the course and some of them did try to test their own code after initially watching the animations. However, the majority of the students who

did not test their own code may have had inadequate time in the laboratory to complete their use of DIVTIC. As the course progressed, the complexity of the programming concepts increased so that the use of DIVTIC for problem-solving was time consuming.

- **Conclusion to strategy in using DIVTIC**

In conclusion to these discussions of the strategy the students used in DIVTIC, the patterns appeared to show that the students tended to interact with the animation and they seemed likely to watch the animation process more and more as the course progressed. Students with a low GPA appeared to use the control buttons to interact with DIVTIC more than others at the beginning of the course, while the students with a high GPA appeared to have more interactions toward the end of the course. On the other hand, students with an average GPA appeared to have more discussion with their peers than others. A minority of the students had a chance to test their own C source code right after watching the animation. The results revealed that the weekly tasks for the students, which were to be completed within a specific time in the laboratory, were likely to be too difficult. In addition, DIVTIC was too time consuming. Students seemed to take more time to use DIVTIC for more difficult tasks. Thus, they seemed to need more time for utilising DIVTIC in such a way that they would perceive and construct their learning processes. However, the animation appeared to be useful in helping students discover and apply information they perceived by constructing their own meaning and knowledge from the animation process.

6.2.3 Animation Component Usage Patterns

There were four different panels in each animation including C Source Code, Message Board, Monitor Output, and Memory Map panels. Interviewees were asked to indicate the most useful panel and the results are shown in Table 6.14.

Table 6.14: The most useful panel of the animation

Most Useful Panel (n = 24)				All Useful Panels
C Source Code	Message Board	Monitor Output	Memory Map	
7 (29.17%)	9 (37.5%)	2 (8.33%)	6 (25%)	17 (70.83%)

There were seven students (29.17 percent) who responded that the C Source Code panel was the most useful for them as one of them stated, *"If we don't understand this part,*

we cannot resolve the question. We should be able to read it before we analyse it" (Student 639); nine students (37.5 percent) responded for the Message Board panel; two students (8.3 percent) responded for the Monitor Output panel; and six students (25 percent) responded for the Memory Map panel. There were seven students (29.17 percent) who responded that all panels were equally useful to them:

> *Student 817: I think they're all important.*
>
> *Student 042: They're equally important as they work together.*
>
> *Student 413: They are equally important. Each part has its own usefulness which shows its functions and tells us what is happening.*
>
> *Student 866: There's no 'the most important' part. They are all important.*
>
> *Student 936: I think they're all important. If we don't have the program, we won't know how it works. We have a Message Board to show us how programs run. Monitor helps us understand the result and Memory Map shows us where to store data. They all make me understand better.*
>
> *Student 121: All parts are important as they relate to each other. They make me understand how the program works and understand the procedures of value and memory storage.*

The results suggest that the Message Board panel was the most important panel as it was the most useful panel to which the students responded. The students appeared to spend most of their time on the Message Board as a strategy in their learning process along with the use of the C Source Code panel. The Monitor Output and Memory Map panels also seemed to be useful options that the students could use in helping their understanding in some cases, as one student claimed, "*This is important sometime. Suppose only once there's a question asking how many bytes it will take to declare the variables, This will help with the answer but after that it is useless" (Student 872).*

Students had different strategies for using DIVTIC. For example, one student used DIVTIC as, *"I finished the question before I checked the answer" (Student 817),* while another used as "*... sometimes if I don't understand the question I look at the answer key before I start: (student 425).* Most students seemed to have a difficult time understanding the problems and solving all of them. One student described his strategy in using DIVTIC as, *"I usually look at the program to see how much I understand it. If I already understand, I won't click 'PLAY', but if I don't understand, I will click 'PLAY' and study" (Student 683).* This appeared to suggest that the use of DIVTIC would help students understand the algorithms.

The following figure (Figure 6.8) shows the percentage students associated with the most useful panel in the animation activities.

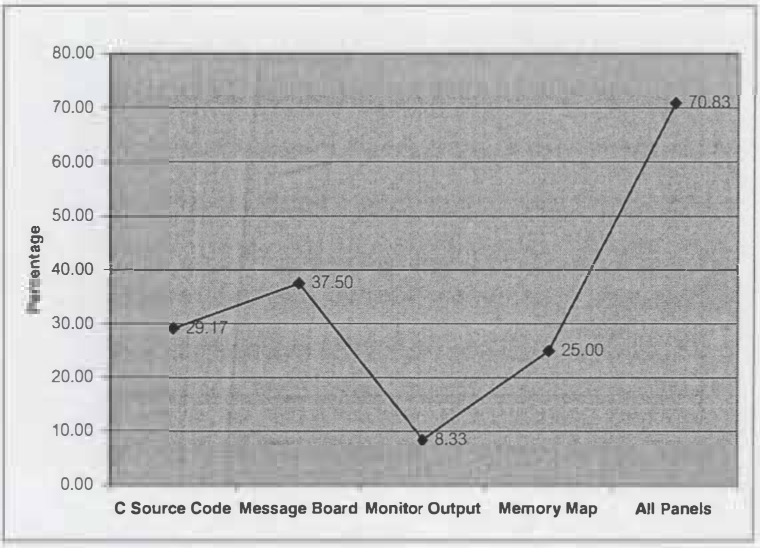


Figure 6.8: The most useful panel in the animation examples component

- **Conclusion to animation component usage pattern**

The results suggest that the Message Board panel was the most important since it was the most useful panel to which the students responded. The students appeared to use most of their time on the Message Board as a strategy in their learning process along with the use of the C Source Code panel. Students had different strategies for using DIVTIC such as finishing the weekly task first and looking for the answers from DIVTIC or vice versa. However, the majority of students (70.83 percent) claimed that all panels in the animation were useful and needed to be linked together within such animation.

6.2.4 Conclusions

The findings suggest that students had fewer interactions with the use of DIVTIC toward the end of the study but more thinking along with the animation process. Many students with a high GPA used the control buttons (e.g., Stop, Play, Backward, and Forward buttons) the most to interact with the animation process throughout the entire course while the students with a low GPA used it less. In addition, the students with a high GPA appeared to use the control buttons more as the course progressed. Some possible explanations for the patterns of usage observed were that students with a low GPA may have made slow progress in learning so that they had no time to use the

control buttons to reflect their thoughts along the animation process. Students with an average or high GPA may have possessed more understanding of programming concepts than students with a low GPA. As the course progressed, the complexity of the programming concepts may have made proved a challenge to the students with a high GPA so that these students tended to use the control buttons more toward the end of the course.

DIVTIC was more likely to engage students when they were in the laboratory session and when they were required to use DIVTIC along with the weekly tasks. About 50 percent of the students seemed to have no time to use DIVTIC when they had free time. Students with an average GPA appeared to use DIVTIC most when they had free time, were faced with programming problems, were doing the assignment, and were staying with friends while students with a low GPA appeared to use DIVTIC most before the examination. Within each animation, the Message Board panel was the most important panel to be concerned with since it was the most useful panel to which the students responded. The students appeared to make most of the Message Board as a strategy in their learning process along with the use of the C Source Code panel. However, the majority of students claimed that all panels in the animation were useful and needed to be linked together within such animation.

6.3 Question 1c: What Factors Influence Students' Use of DIVTIC?

The methods used to collect data to answer this question included (a) DIVTIC self-administered evaluation questionnaires, (b) Subject semi-structured interview; (c) Tutors' observations; (d) Researcher's observations; (e) Tutor semi-structured interviews, and (f) Screen recordings. The use of each method is described as follows:

- **The DIVTIC self-administered evaluation questionnaire form:** Only part 3 in this form was used in weeks 6 and 10 asking the students to describe the problem in relation to DIVTIC included software, hardware, network, and other.
- **Subject semi-structured interview:** There were 24 interviews in total throughout the study. Each interview took approximately 5 minutes. A semi-structured interview format was used, which was comprised of five questions

to, explore how the Animated Example component help students to understand algorithms as follows:

- Feedback on how many problems students watched?
- Did students finish all problems in the weekly task?
- Ask some questions about problems itself, e.g., ask about the algorithm to see if students actually learned?
- Which part of the animation helped students learn the most and why?
- What should be improved to make DIVTIC a better tool?

The summary of all 24 interviews was categorised into six patterns: (a) Finished all problems, (b) Useful/Interesting, (c) Difficulty in language, (d) Most useful part, (e) Understanding the problems, and (f) Suggestions. However, only pattern (d), the most useful part pattern, was used for exploring this question.

- **Tutors observation:** Tutors observed the laboratory session from weeks 2 to 12 except weeks 7 and 11, which was when the laboratory tests 1 and 2 were taken. The Tutor observation form was comprised of three open-ended questions as follows:
  - What questions did students ask when using DIVTIC?
  - What help did you need to give to students?
  - What problems did you face in this week relating to DIVTIC (Hardware, Software, Network, and Other)?
- **Researcher's observation:** The researcher observed the laboratory session focussing on the problems that occurred during each session from weeks 2 to 12, except weeks 7 and 10 in which the laboratory test 1 and 2 were taken, respectively. The problems were divided into 4 open-end questions: (a) Hardware, (b) Software, (c) Network, and (d) Other.
- **Tutor semi-structured interviews:** There were two tutors to interview namely Tutor A and Tutor B. The transcripts of the interview were translated into English and divided into two phases, the first interview and second interview. Both interviews were designed to be taken during the laboratory tests 1 and 2, respectively, outside the laboratory room itself. Each interview

with each tutor took approximately 10 minutes. The semi-structured interview format was used and was comprised of five questions as follows:

- What did you think about Animated Examples e.g., interface, useability, clarity, user-friendliness, and value?
- As a tutor, did you like Animated Examples, how and why?
- From the students' perspective, did you like Animated Examples, how and why?
- What other features did you think Animated Examples should have?
- Did you have any other comments about Animated Examples?

- **Screen recordings:** Screen video capture software was used to record the activities of three voluntary students using DIVTIC for approximately 30 minutes each week starting from weeks 3 extending through to 12, excluding weeks 7 and 11, in which the laboratory tests 1 and 2 were given, respectively. In total, there were 24 screen recordings, including five students with low GPA, seven students with average GPA, and 12 students with high GPA. These screen recordings were used to explore which components in DIVTIC students used and for how long. At the beginning of each laboratory, three students were asked to volunteer to use the screen video capture software to record their screen while they were using DIVTIC. The weekly task was also given to the students at the beginning of the laboratory. Therefore, students needed approximately 10 to 15 minutes to complete the weekly task and then they would log into the DIVTIC system to play the relevant animation for the weekly task. This was expected to last from 15 to 20 minutes.

There appeared to be some factors in DIVTIC's usage that seemed to have potential to encourage or impede students' use of DIVTIC. These included technical factors, content factors, and factors to do with the design of DIVTIC. The statistics software application, SPSS, was used to calculate frequency distributions.

6.3.1 Technical Factors

The technical factors in this study that may have impacted on student learning were based on hardware, software, and network problems. The following table (Table 6.15)

shows the problems that occurred during the laboratory session and outside the laboratory as well.

Table 6.15: Forms of problem in using DIVTIC (Weeks 6 and 10)

Topic	Week 6 (N = 39, Missing = 8)		Week 10 (N = 39, Missing = 8)	
	No	Yes	No	Yes
Hardware problem	30 (96.8%)	1 (3.2%)	31 (100%)	0 (0%)
Software problem	30 (96.8%)	1 (3.2%)	30 (96.8%)	1 (3.2%)
Network problem	27 (87.1%)	4 (12.9%)	28 (90.3%)	3 (9.7%)
Other	31 (100%)	0 (0%)	29 (93.5%)	2 (6.5%)

In week 6, there was only one student (3.2 percent) who reported having a problem with the hardware and another who reported having a problem with the software. They said it worked slowly. However, the majority of the students, 30 students (96.8 percent), reported not having any problems with hardware and software configurations. There were four students (12.9 percent) who reported having a problem with the network among whom two could not connect to the web site and the other two said that the network was too slow.

In week 10, there were no hardware problems reported. There was only one student (3.2 percent) who reported having a problem with the software and three students (9.7 percent) who reported having a problem with the network saying that it was too slow. There were two students (6.5 percent) who stated that the text in the message board was too small and difficult to understand as it was written in English.

The results seem to suggest that the majority of the students was satisfied with the DIVTIC setting. There was only one student who stated that the explanation message in the Message Board panel should be written in Thai. This revealed that there were few factors impeding students' use of DIVTIC with regard to software, hardware, and network problems as indicated by the lack of students who reported having a problem with the setting.

In order to access the DIVTIC system, students needed to use computers that already had Internet-ready connections. Students seemed to have problems with Internet connections when working outside the laboratory. As Tutor B stated *"Students find it hard to keep up with the unprepared computer laboratory and problems with web sites and Internet connections."* He suggested that DIVTIC could be used without the

Internet connection requirement if DIVTIC was copied onto CDs and distributed to all students. Students seemed likely to prefer using DIVTIC in their own time outside the laboratory. He also claimed "*Students can then use it anytime without accessing the Internet. This will help them learn.*"

Tutors also indicated that the network connection was very slow and sometimes students could not connect to the server because of an error on the proxy server. From Tutor B's experience in setting up the web site for courses, he claimed that students seemed to be unwilling to do such a course via the Internet. Thus, he suggested that DIVTIC could be set up for use at home:

> *... copy it onto a CD for students. They can use it at home instead of going to the web site each time when they want to access it. I understand that this may need to be researched and developed then amended. This all takes time. Once it's complete we can copy it onto a CD. Students can then use it anytime without accessing the Internet. This will help them learn... This is an idea of how students will use the program without accessing the Internet.*

The results from the researcher's observations also showed that the network problem was a major problem that influenced the use of DIVTIC as the network connection at SUT was not estabilised. Any other problems could be resolved during the laboratory session such as some typing errors in the weekly task problem, or some mismatches between images and their given descriptions in the animated example component.

6.3.2 Content Factors

There were two content factors that emerged from the students' interviews that seemed to have a capacity to influence the students' use of DIVTIC. These related to the difficulty of the problems in the weekly task and the language used in DIVTIC. The following table (Table 6.16) shows the percentage of students who completed all problems in the previous two weeks, completed it sometime, or never completed it at all.

Table 6.16: Difficulty of problem and influence on students' use of DIVTIC

Finish all Problems		
Yes	**Sometime**	**Never**
7 (29.17%)	11 (45.83%)	6 (25%)

There were seven students (29.17 percent) who indicated that they finished all problems in the previous two weeks. Not all of them were sure that they got all correct answers.

One of them argued, *"I think there are some mistakes" (Student 042)* and another argued, *"... but not sure that they're correct" (Student 413).* There were 11 students (45.83 percent) who finished all problems at sometime while there were six students (25 percent) who never finished any of them. The students who sometimes finished and who had never finished all had problems:

> *Student 626: I could do only half of it. It's also very difficult.*
>
> *Student 572: I usually finish Question 1. Question 2 I only finish half...*
>
> *Student 941: I finished them in the few first weeks, but later it started to get harder and more complicated so I haven't finished them all.*
>
> *Student 091: Sometimes I can finish but sometimes I can't because the time's up.*
>
> *Student 936: No. Sometime I can finish them but sometimes I can't. If I have enough time I may finish them.*

Some interviewees were asked some algorithm questions relating to the weekly problems they had done. Most of them appeared to misunderstand the programming concepts. For example, one of them could not answer the question "Given: ***int N[5] = {15, 22, 3, 55, 42}***; what is ***N***?" He could not answer this simple question correctly. He said that ***N*** was comprised of five members. The correct answer was that ***N*** was an array containing five integer variables. Another example from weekly task 7 (Pointers) was that students misunderstood how to define a pointer variable. When a question "Given: ***int *ip;*** what is ****ip***?" was asked. A student replied that ****ip*** was a function. This was the incorrect answer. The correct answer was that ****ip*** was a pointer variable that could point to any integer variable. Some of interviewees understood the problems but they were not able to complete them with all correct answers. This indicated that students had difficulty in understanding the programming concepts and algorithms.

The problems in the weekly tasks seemed to be difficult for the students to complete fully in the given time. Some students seemed to have a difficult time understanding the problems. Using DIVTIC was time consuming and the given time to complete the weekly task seemed inadequate. The difficulty of the weekly task and the inadequate time given may have discouraged students from using DIVTIC.

Another factor that appeared to impede students from using DIVTIC was the language. DIVTIC was designed to use English to explain the programming process in the Message Board panel located in the animated example component. The following table

(Table 6.17) shows the percentage of students who preferred to have the explanation message in the Message Board panel be in Thai rather than in English because they seemed to have difficulties in learning English at the same time as the programming concepts. This seemed to be a factor that misled their understanding.

Table 6.17: Language influences on students' use of DIVTIC

Message Board Panel in Thai	
Yes	**No**
20 (83.33%)	4 (16.67%)

There were 20 students (83.33 percent) who replied that the English explanation was an obstruction to their learning. They responded to this English explanation in the following ways.

> *Student 817: It would be good if we can get it in Thai.*
>
> *Student 425: I think it would be good as this would help us understand better.*
>
> *Student 369: I wish it were in Thai language because I find it hard to understand some words.*
>
> *Student 413: Yes, it should be in Thai language... not every one of us knows English well.*
>
> *Student 936: Yes, that's very good as some people may lose interest if they don't understand English.*

However, there were four students (16.67 percent) who replied that they still wanted to have the explanation in the Message Board in English as they argued

> *Student 566: Yes, I think it's good as we can practise English... I think sometimes it's not appropriate because we will lose the meaning.*
>
> *Student 479: I can understand English both reading and interpreting so there's no problem for me. But for other students, who don't understand English, they need to see it in Thai.*
>
> *Student 091: I still want it to be in English as we will learn lots of vocabularies.*
>
> *Student 941: It looks international and some words can be transliterated. Some words cannot be translated into Thai as they will lose their meanings.*

The percentage of the number of students who wanted to have Thai explanation was significantly higher (83.33 percent) than those who wanted an English explanation (16.67 percent). This shows that the difficulty in understanding English appeared to be the most obstructive factor for Thai students since English is a foreign language to them.

DIVTIC was designed to be a supplementary tool to help students learn. It provided a Message Board panel that displayed related information as the animation progressed. The information was written in English, which both tutors agreed, may have been a problem for students. For example, Tutor A pointed out that students could *"misinterpret or misunderstand"* and Tutor B stated that *"These students will try hard to study English at the same time."* Students seemed to grow tired of learning both new English vocabularies and programming concepts at the same time. They were not always able to achieve both learning targets. The language difficulties and the complexity of the concepts seemed to be a cause of students having negative attitudes towards DIVTIC such as feeling of boredom, confusion, diffidence, or even discarding DIVTIC altogether.

The tutors also pointed out the clarity and user-friendliness of DIVTIC as follows:

> *The program is quite clear and helps in less complicated matters. For example, at the printf or scanf functions, the program shows calculation step by step. The Pointer and Function are complicated, students may need to ask a tutor. But it is better for them to work and practise on this program rather than not seeing it at all.*

This reply also appeared to support the notion that students faced with less complex concepts would have fewer questions to ask tutors. On the other hand, more questions would be asked when concepts were more complex. This appeared to be a factor that could encourage students to use DIVTIC, if it was used with topics that were less complex where the step-by-step visualisation was easy to follow and understand. However, it may deter students from using DIVTIC when the concepts were complicated or inadequate information was provided for a topic.

The following table (Table 6.18) shows the summary of the problems that occurred, as observed by the tutors, of the concept and weekly task.

Table 6.18: Summary of tutors' observation

Problem	Detail
Concept	• Byte representation in memory? • What is *sum += x;*? • *for(i=1; i<=3; i++);* *i++* and *++i* are the same? • What *LuckyNumber( )* does? • How do we know the value of *n*? • What is *strlen( )*? • What is *int backward(char[ ]);*? • Where the returned value should be located? • What is **p = n;* and **p = &n;*? • What is *sizeof(s)*?
Weekly Task	• Did not understand how the DIVTIC Weekly Task work? • In OPERATORS, student did not understand what had been asked for in problem 2, question 1. • What do they need to do? • In FUNCTION, problem 2, Q4, change *j = 6;* to *i = 6;*

Most of the problems that occurred during the tutors' observation were concerned with the concepts of the C programming language. The students asked more questions about conceptual problems as the materials got more complicated. This problem could effect the students' use of DIVTIC if they did not understand the concept behind the materials. They might either stop using it altogether or rarely use it. However, DIVTIC seemed to be a useful tool for helping students learn some basic concepts to start with, as shown by the time spent using DIVTIC from log files as discussed earlier.

The weekly task problem could also have influenced students' use of DIVTIC. If the students understood the weekly task questions and could solve it, they would use DIVTIC to check their answers. This ensured that they understood the material. On the other hand, if the weekly task was too complicated and unclear to them, then they could feel that DIVTIC was a boring tool and not invest time in using it.

In addition, from the screen recording of 24 voluntary students, there was some evidence that the difficulty of the problems in weekly task itself and the language seemed to discourage students' use of DIVTIC. The following table (Table 6.19) shows the number of students who watched the animation for one or two or more times and who used the Thai dictionary program to help them understand some of the vocab.

Table 6.19: Number of students who watched animation one or more times and used Thai dictionary program

Number of the students watched the animation		Number of the students used Thai dictionary program
One Time	**Two or More Times**	
8 (33.33%)	16 (66.66%)	3 (12.5%)

Table 6.19 shows that there were eight students (33.33 percent) who reported playing and watching the animation only one time while 16 students (66.66 percent) reported playing and watching the animation for two or more times. This seemed to indicate that there were two thirds of the students who seemed to have difficulty in understanding the programming concepts or in solving the weekly task. The difficulty of the programming concepts or of the weekly task could discourage students' use of DIVTIC. Another influencing factor seemed to be the difficulty of the language itself. The explanation in the Message Board panel of the animation was written in English. There were three students (12.5 percent) who used the Thai dictionary software while they were using DIVTIC. They seemed to have difficulty in understanding some of the vocab. This

could have affected the students' use of DIVTIC and also have discouraged the students from using DIVTIC.

6.3.3 The Design of DIVTIC Factors

The design of DIVTIC could also be a factor that influenced students' use of the tool. The following tables show the summary of some feedback from both Tutors A and B. This feedback provided insight into some of the factors relating to the design of DIVTIC that appeared to influence the students' use of DIVTIC. The first interview with Tutors A and B was taken in week 7 and is summarised as follows:

Table 6.20: First interview with Tutor A

Question	Tutor's Feedback
Interface	• Interesting, good design, easy to use and access
Useability	• Most of the time, students can access and understand on their own, Less questions to ask tutors during the lab session
Clarity	• Good and clear for basic and simple commands, Pointer and Function are advance and difficult tasks in which students may need to ask tutors
User-friendliness	• Easy to access with choices of menu to navigate
Value	• 7 to 8 because of dislike of the colour of the design
As a tutor/instructor, do you like Animated Examples, how and why?	• Like: -- It is beneficial for the students -- Save a lot of time in answering some basic programming problems • Dislike: -- DIVTIC can replace his duty
From the students' perspective, do you like Animated Examples, how and why?	• Love DIVTIC • New media and can replace the textbook
What other features do you think Animated Examples should have?	• Enough information and features • No need to include more images or other features
Other comments about Animated Examples?	• Having DIVTIC is better than not having • Get more understanding from DIVTIC than Textbook • Should be in Thai Language

Table 6.20 reveals that Tutor A appeared to be satisfied with all DIVTIC elements. His responses appeared to support the notion that the interface and useability of DIVTIC could be factors that influenced students to use DIVTIC:

> *The Interface is interesting and has a Menu usage. It is easy to understand and easy to go back to the Menu. It can be accessed easily and quickly. I like it at this point. When I use the program, because I already understand the C language it seems to be easy for me. I notice that between the students who have never used the program before and ones who have used it, their questions are different. Students who have no knowledge of this will ask questions frequently. For experienced students, they will study it themselves and hardly ask any question. The program helps them understand and resolve problems for them.*

Another point he made is that DIVTIC could be used as a supplementary tool for learning C language since it was a self-learning package that students could study and understand by themselves:

> *If I were a student, definitely I would have many books about the C program. If I saw media like this, I would like it. I wouldn't have to have a text book all the time as I would be able to study through the media. I can look up the Help function when I get stuck. And if I don't understand because it's in English, I can then go to the DIVTIC program and run the program to help me resolve my question. I could study by myself and get most of it done.*

By using DIVTIC as a supplementary tool, he claimed that students would gain some benefits out of it. He rated DIVTIC as 7.5 out of a scale of 10. He appeared to like all the elements of the DIVTIC system, except for the colour, *"If it's my colour, which is based on Blue, I'll give you 10 out of 10"*. However, from the tutors' perspective, he confessed that his position as teacher could be replaced by the DIVTIC system if it was fully developed.

> *in regard to being a tutor, I worry that I may lose my job as this program will replace me. It is possible that students won't need a tutor because they hardly ask in class now. In future, the institute may consider cutting down tutors and leave only a few for students' support. This is a personal worry, that there is an advantage to having the program and a disadvantage to a tutor.*

Table 6.21: First interview with Tutor B

Question	Tutor's Feedback
Interface	• Very good
Useability	• Good and help student to imagine and see what is happening
Clarity	• Message Board is needed to be changed into Thai language to make it easy to understand for weak students who have a difficult time to understand English. Sound would be more appropriate and useful than static text in Message Board
User-friendliness	• Easy to access and use
Value	• Now, it is only 7 points. If Message Board is in Thai with sound, then DIVTIC would get 10 points
As a tutor/instructor, do you like Animated Examples, how and why?	• Love -- Step-by step animation -- Help to understand
From the students' perspective, do you like Animated Examples, how and why?	• Like it and use it more if it is in Thai language
What other features do you think Animated Examples should have?	• Now it is good enough • More useful if the users can enter their own input
Other comments about Animated Examples?	• Change Message Board from English to Thai • Add audio explanation • Copy onto CD and handout to students

Table 6.21 reveals that Tutor B appeared to be satisfied with the basic environment of the DIVTIC system. His response when asked about the interface and useability was, *"It's very good. The Animation helps students understand how the program works. It helps with imagination and tells us what's going to happen next. I think this animation component helps students lots"*. The tutor agreed that both the interface and useability

of the DIVTIC system seemed good and appropriate so that the students could follow and visualise what was going on. He commented on the user friendliness of DIVTIC, *"This is already good. Students enjoy using this function. Wherever they click, there will be an explanation telling them what's happening".* However, this tutor suggested that the explanations in English were not always appropriate for the Thai students. The tutor felt that DIVTIC needed some modifications to make it more appropriate for Thai students so that they could learn in their Thai language:

> *There are two issues about this. First, it's in English which may be difficult for Thai students. If it were in Thai language, it would take less time trying to understand the program. Thai people will find it hard to read and understand. Second, in the Animation there is reading and listening. It would be better if there could be sound at the text explanation. Students would understand better.*

Overall, the tutor agreed that the overall design which included interface, useability, clarity, and user friendliness would influence students' use of the tool even more so if it were designed with a good look and feel and understandable information. He ranked the value of DIVTIC as 7 out of a scale of 10 because he insisted that the, *"... Thai language is necessary, so is sound. If you have both of them, I'll rate 10 out of 10. For now, I think 7 is fair to rate".*

From a student' perspective, the tutor claimed, *"I like it as this is a main part of studying. Students cannot imagine what's going to happen next and if we have the Animation to help them, this will be the best point of the program."* His response supports the important role of visuals in helping learners to perceive abstract concepts. On the other hand, from his impression of the students' perspective, he added,

> *I don't think I can answer this question very well but I'll try to answer it fairly. My English is okay comparing to students who don't know the language. These students will try hard to study English at the same time. For example, some students that their study require English text books and they ask whether they can avoid using them. This is their problem with vocabulary learning. There are also students that they need to study via a web site. Students are reluctant to do so as they're not familiar with the language. I have collected statistics about this. You can take a look later. Students find it hard to keep up with the unprepared computer laboratory and problems with the web site and Internet connections. In regards to the program's benefits, I think it is good and will be beneficial. If the text were in Thai language with Thai sound, this would make it even better.*

The tutor insisted that the language was the most important influential factor impeding learning effectiveness. Since this tool was designed for Thai students, the language should have been in Thai to avoid misunderstanding.

In pointing to the animated examples, the tutor appeared to be satisfied with all the features provided in the DIVTIC system. In addition, he suggested that DIVTIC might provide the extra option for students to test themselves:

> *But I would like to see students inputting their own program and run it. The program will then show them what is happening at each step. The Animation will then help students to achieve their Ultimate goal. The current program has a pre-set Fix Input which corresponds to the course outline. I think this is okay, nothing to lose. But I still need to see students input their own program.*

The second interview with Tutors A and B was undertaken in week 11 during the laboratory test 2 and is summarised in Table 6.22.

Table 6.22: Second interview with Tutor A

Question	Tutor's Feedback
Interface	• Interesting; good design; easy to use and access; Like this point
Useability	• Easy to understand the content; Good organisation of the content
Clarity	• May need to convert into Thai language, since it is used to teach Thai students to avoid misunderstanding or mistranslation.
User-friendliness	• Easy to access; Choices of menu to navigate
Value	• 7 to 8 because of dislike the colour of the design
As a tutor/instructor, do you like Animated Examples, how and why?	• Surprise in getting into the content.
From the students' perspective, do you like Animated Examples, how and why?	• Enjoyable • Just need a click on it and it will run automatically • Excellent tool
What other features do you think Animated Examples should have?	• Message Board needs to be changed into Thai Language • May use speech instead of the static text
Other comments about Animated Examples?	• Colours are also needed to be changed

Table 6.22 reveals that Tutor A seemed to be satisfied with the interface, useability and user friendliness of DIVTIC:

> *It is easy to understand and easy to retrieve back to the Menu. It's quite good. It tells us how to operate it in order for it to be easy to understand. This means the user only has to click each button to explore quick details and information. It's a user friendly program and matches our needs. I feel the program's ability is just right.*

However, the tutor pointed out that explanations in English needed to be considered. He argued, *"This can be a barrier for Thai students as they don't understand specific words. This may cause misinterpretation or misunderstanding"*. He still insisted he gave 7.5 out of 10 because he disliked the colours of the design.

From the tutors' perspective, he replied, *"In my opinion, direct access to details is amazing... This animation will attract students because it makes the program more interesting... This makes students alert and won't be bored by 'clicking and waiting'"*.

He seemed to be surprised with the access to content. On the other hand, from his observations of the students, he seemed to be pleased to use DIVTIC and indicated a willingness to continue using it as a tool to aid his students' learning:

> *Certainly, if I were a student I would enjoy using the animated examples as they would be important to me. I would just move the cursor and each time I moved it, something would happen with an explanation. This is like something that has already been prepared for users for easy learning. I would like it very much if I were a student.*

The tutor also suggested that some features might need to be changed including the explanations in the Message Board panel in which English text should be replaced with Thai. Furthermore, he suggested adding some audio explanations to make it easier for the learners, as he stated, *"It would be good if the English explanation were in Thai. This may be difficult I know as the program will not be used in Thailand only. Another suggestion is the colour".*

Table 6.23: Second interview with Tutor B

Question	Tutor's Feedback
Interface	• Good
Useability	• Text in Message Board is too small
Clarity	• It is okay.
User-friendliness	• It is an excellent toll that is easy to navigate. Students went back and forth easily around the toll.
Value	• 7 or 8, because it is needed to have some changes, e.g., Thai language, audio explanation etc.
As a tutor/instructor, do you like Animated Examples, how and why?	• Like -- can use the tool to animate some examples in the lecture room -- help student to visual what is happening inside of the computer
From the students' perspective, do you like Animated Examples, how and why?	• Like -- show what is happening inside the computer
What other features do you think Animated Examples should have?	• Audio explanation instead of the statice text explanation • Have an option for students to put their own input or source code and let the software create the animation and show them visually how it works
Other comments about Animated Examples?	• Good enough • More useful if the users can enter their own input.

Table 6.23 reveals that Tutor B was also satisfied with the interface and clarity. He stated, *"I think the Interface is interesting. Students can imagine what will happen when they run the program. The animation helps them understand the memory better"* and *" The clarity is alright. Also, when running the program, we can imagine what's going to happen next".* However, the tutor indicated a problem with the useability of DIVTIC in that the text explanation was too small and he suggested a solution to make it more useable:

> *There is a small problem about the Usability - the text is too small and hard to read. It's in English and I'm not sure students will be able to cope with the*

> *new technique of teaching and learning style. They hardly access the web site. And if we copied it on a CD, again, I couldn't say they would play it. I know that if we let them access the web site, they will use their computers at the dormitory which I can't check. I can only look at the Log File to check their usage.*

In term of user friendliness, DIVTIC seemed to be easy to navigate through, as the tutor stated, *"It is certainly easy for students - only clicking buttons. They don't need to have background. I noticed students were confident in clicking and exploring it".*

Tutor B preferred to have DIVTIC as an additional tool because of its usefulness. In support of this he said,

> *It would be good if I could use this in my teaching. This will help me explain to students what's happening. At present I'm not able to show them the Animation. It would be beneficial to them if they could see what was going on in the computer. This could be a problem to them if they couldn't see it.*

From his impression of the students' perspective, he strongly stated that he liked it because of the visual as the traditional teaching style does not provide animation:

> *I think I would like it. At least it would tell me things that a tutor couldn't tell in class. This function will show me how a variable happened and where a memory came from. And when I click a 'run' command I can watch out for a result. I can see what's going on and can use the information for other variables. I can also see how the Loop runs each time. If I were a student I would definitely like it.*

The tutor rated the value of DIVTIC as 7.5 on a scale out of 10. He claimed that DIVTIC needed some additional features including audio explanations and explanations in the Thai language. He also suggested to add an ultimate goal to the DIVTIC system by providing an option for students to put their own input or source code to let the software create the animation and show them visually how it works.

The following table (Table 6.24) shows factors that tutors suggested would influence students' use of DIVTIC in two phases of interview, the first interview and the second interview. Table 6.24 shows encouraging factors, while Table 6.25 shows discouraging factors. Each factor was rated by using a five-point Likert rating scale. A rating of 1 indicated less influence to students' use of DIVTIC where a rating of 5 indicated the most influence to students' use of DIVTIC. However, a horizontal block of unselected numbers in any column for each tutor indicated that there was no feedback about that factor.

Table 6.24: A summary of encouraging factors from the two phases of interview

Encouraging Factor	First Interview		Second Interview	
	Tutor A	Tutor B	Tutor A	Tutor B
Interface	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤
Useability	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤
Clarity	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤
User-friendliness	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤
Value	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤
Thai explanation	① ② ③ ④ ❺	① ② ③ ④ ❺	① ② ③ ④ ❺	① ② ③ ④ ❺
Audio explanation	① ② ③ ④ ⑤	① ② ③ ④ ❺	① ② ③ ④ ❺	① ② ③ ④ ❺
Testable their own input	① ② ③ ④ ⑤	① ② ③ ④ ❺	① ② ③ ④ ⑤	① ② ③ ④ ❺
CD version beside Web-based	① ② ③ ④ ⑤	① ② ❸ ④ ⑤	① ② ③ ④ ⑤	① ② ❸ ④ ⑤

Table 6.24 shows that both tutors agreed in both interviews that the interface, useability, clarity, and user-friendliness were very important factors. The design of DIVTIC and its context had played a major role that could influence students to use DIVTIC either more or less. A good design and well organised components with sufficient information would engage students to use it more: *"Tutor B: It is certainly easy for students—only clicking buttons. They don't need to have background. I noticed students were confident in clicking and exploring it"*. On the other hand, students would appear to use it less if such a tool had an inappropriate design or insufficient information in its context.

Both tutors suggested changing the English explanations in the Message Board panel to Thai with the inclusion of an audio explanations instead of using static text: *'Tutor B: Instead of just a cursor moving, there should be a sound explanation to help students understand better...If the text were in Thai language with Thai sound, this would make it even better."* These suggestions appeared to be the most important suggestions to encourage students to use DIVTIC as Tutor B claimed, *"... it would take less time trying to understand the program."*

Tutor B suggested adding a feature to DIVTIC which would enable students to test their own input or source code and let DIVTIC automatically create an animation for them. This suggestion seemed to be the most powerful option in encouraging student to use DIVTIC. Tutor B also suggested copying DIVTIC onto CDs and distributing it to all students to avoid facing problems with the Internet connection so that students could use it in any place at any time without having any Internet connection concerns, as he stated, *"We can then copy it onto a CD for students. They can use it at home instead of going to the web site each time when they want to access it"*.

This suggestion appeared to provide ways to encourage students to use DIVTIC since they could have their private time to concentrate more on the learning process via the use of DIVTIC. However, this suggestion would only provide an option for students to use it at their own pace and in their own time. They would also need to use it in the laboratory with some allocated limited time to ensure that students did use DIVTIC as Tutor B explained *"... if we copied it on a CD, again, I couldn't say they would play it."*

The following table (Table 6.25) shows factors that appeared to discourage students' use of DIVTIC. There were four factors from the tutors' feedback including the complexity of the concepts, the English explanations, the time it took, and the Internet connection.

Table 6.25: A summary of discouragement factors from the two phases of interview

Discouraging Factor	First Interview		Second Interview	
	Tutor A	Tutor B	Tutor A	Tutor B
Complexity in concepts	❶ ② ③ ④ ⑤	① ② ③ ④ ⑤	① ② ③ ④ ⑤	① ② ③ ④ ⑤
English explanation	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤	① ② ③ ❹ ⑤
Time consuming	① ② ❸ ④ ⑤	① ② ③ ④ ⑤	① ② ③ ④ ⑤	① ② ❸ ④ ⑤
Internet connection	① ② ③ ④ ⑤	❶ ② ③ ④ ⑤	① ② ③ ④ ⑤	❶ ② ③ ④ ⑤

The difficulty of the problems could be a factor which discouraged students from using DIVTIC. However, this had little overall effect since DIVTIC still seemed to be a useful tool for students as they could see the animations of any difficult problems. Tutor A stated in the first interview, *"The Pointer and Function are complicated, students may need to ask a tutor. But it is better for them to work and practise on this program rather than not seeing it at all".*

The English explanations in the Message Board appeared to impede students from using DIVTIC since their native language was Thai: *"Tutor A: This can be a barrier for Thai students as they don't understand specific words. This may cause misinterpretation or misunderstanding".* Both tutors suggested replacing the English explanations with Thai explanations or to have both English and Thai versions in DIVTIC so that students would have a choice of selecting a version that was more appropriate for them. This could be a major factor influencing the use of DIVTIC. As Tutor B stated, *"It's in English and I'm not sure students will be able to cope... "*

Time also could have been a factor that discouraged students from using DIVTIC. Students were sometime allowed to spend up to one hour with DIVTIC and other times they were allowed only 30 minutes depending on the level of difficulty of the task. The average time spent using DIVTIC in the laboratory was approximately 45 minutes for the entire study. This seemed to be inadequate time for students to complete the whole task. Tutor A commented, *"For the first hour, I think it should not be limited to this hour only... I don't think it can be predicted. This is dependent upon each student."* Tutor B also added, *"Students may need lots of concentration in class to be able to understand. It takes time."*

Another factor that seemed to discourage the use of DIVTIC was the Internet connection. Tutor B explained from his own experience, *"Students find it hard to keep up with the unprepared computer laboratory and problems with web sites and Internet connection."* This factor, however, had little effect on the use of DIVTIC in the laboratory since the set up of the computer network ran successfully and the technicians were always on board and ready to solve any unexpected problems. Difficulty in connecting to the web site happened only once and it took several minutes to fix it.

6.3.4 Conclusions

The findings suggested that the majority of the students seemed to be satisfied with the DIVTIC setting. However, if DIVTIC could have been put onto CDs and given out to all students, they would have been able to use it at their own pace and in their own time without any Internet connection problems. The problems with the language used in the Message Board panel of the animated examples component will be further discussed in the next section.

The difficulty of the weekly task and the English explanation in the Message Board panel of the animated example component seemed to be strong factors in obstructing students' use of DIVTIC. Some possible explanations of this may have come from the fact that learning to program is a complicated task and English language is also difficult for Thai students as it is a foreign language. When the students had to learn both programming concepts and English at the same time, they understandably were not to be able to handle both.

In most cases, each aspect of DIVTIC seemed to encourage students to use DIVTIC. The language problem seemed to be a major discouraging factor. To overcome this problem, DIVTIC could be modified by changing English explanations into Thai. All changes suggested in the interviews are shown in Table 6.26.

Table 6.26: Summary of the interviewees' suggestion from students and tutors

Interviewees' Suggestion
• English explanation should be in Thai • The animation run a little bit too fast • Should be able to test their own input • Should add audio explanation • Text should be a little bit bigger • Should be able to click where they want to run • Need more questions in each week to practice

Another major factor that could impede students' use of DIVTIC appeared to be the difficulty of the content and the time consuming nature of DIVTIC. Using DIVTIC was time consuming and the time given to complete the weekly task seemed to be inadequate. The difficulty of the weekly task and the little allocated time could have discouraged students from using DIVTIC. To improve DIVTIC to be more relevant and dynamic, an option to allow students to test their own input is being considered.

6.4 Question 1d: What Attitudes do Students Generate Towards DIVTIC?

The methods used to collect data in answering this question included DIVTIC self-administered evaluation questionnaires, Student semi-structured interviews, Tutor observations, and Tutor semi-structured interviews. The use of each method was described earlier in section 6.3.

- **The DIVTIC self-administered evaluation questionnaire form**: This form was used twice, in weeks 6 and 10. Only the questionnaire which included higher-order thinking, confidence, encouragement, user friendliness, enjoyment, perceived educational value, perceived level of interest, useability, clarity, and collaboration patterns was used to answer this research question.
- **Subject semi-structured interview:** There were 24 interviews in total throughout the study. Each interview took approximately 5 minutes. A semi-structured interview format was used, which was comprised of five questions

to, explore how the Animated Example component help students to understand algorithms as follows:

- Feedback on how many problems students watched?
- Did students finish all problems in the weekly task?
- Ask some questions about problems itself, e.g., ask about the algorithm to see if students actually learned?
- Which part of the animation helped students learn the most and why?
- What should be improved to make DIVTIC a better tool?

The summary of all 24 interviews was categorised into six patterns: (a) Finished all problems, (b) Useful/Interesting, (c) Difficulty in language, (d) Most useful part, (e) Understanding the problems, and (f) Suggestions. However, only pattern (d), the most useful part pattern, was used for exploring this question.

- **Tutors observation:** Tutors observed the laboratory session from weeks 2 to 12 except weeks 7 and 11 in which the laboratory tests 1 and 2 were taken, respectively. The Tutor observation form was comprised of three open-end questions as follows:
    - What questions did students ask when using DIVTIC?
    - What help did you need to give to students? And,
    - What problems did you face in this week relating to DIVTIC (Hardware, Software, Network, and Other)?
- **Tutor semi-structured interview:** There were two tutors to interview namely Tutor A and Tutor B. The transcripts of the interview were translated into English and divided into two phases, the first interview and second interview. Both interviews were designed to be taken during the laboratory tests 1 and 2, respectively, outside the laboratory room itself. Each interview with each tutor took approximately 10 minutes. The semi-structured interview format was used and was comprised of five questions as follows:
    - What did you think about the Animated Examples e.g., interface, useability, clarity, user-friendliness, and value?
    - As a tutor, did you like the Animated Examples, how and why?

- From the students' perspective, did you like the Animated Examples, how and why?
- What other features did you think the Animated Examples should have?
- Did you have any other comments about the Animated Examples?

The following discussion draws conclusions from the students' perspective.

6.4.1 Learning Potential

- **Higher-order thinking**

The questionnaire had a series of questions that sought to explore the way in which using DIVTIC encouraged students to engage in higher-order thinking processes. Table 6.27 shows the questions and the students' responses.

Table 6.27: Higher-order thinking (week 6)

	Higher-order Thinking Week 6	SD	D	NA	A	SA
1.	When I am watching the animation, I stop it from time to time to reflect on what I am trying to get out of it? (N = 39; Missing = 8)	1 (3.2%)	1 (3.2%	6 (19.4%)	17 (58.4%)	6 (19.4%)
2.	Using Animated Examples help me to think logically during the animation process. (N = 39; Missing = 8)	2 (6.5%)	0 (19.4%)	6 (19.4%)	18 (58.1%)	5 (16.1%)
3.	I always discuss with my peers about the animation running process. (N = 39; Missing = 9)	1 (3.3%)	3 (10.0%)	6 (20.0%)	16 (53.3%)	4 (13.3%)

In week 6, Table 6.27 shows that 23 students (77.8 percent) claimed, in both questions 1 and 2, to have stopped the animation and reflect on it while only two students (6.4 percent) never did this. Question 3 explored the ways in which the students discussed DIVTIC with their peers. It shows that 20 students (66.6 percent) claimed to have discussed it with their peers. The results seem to show that DIVTIC was successful in encouraging and supporting this activity which in turn, encouraged higher-order thinking.

Table 6.27a: Higher-order thinking (week 10)

	Higher-order Thinking Week 10	SD	D	NA	A	SA
1.	When I am watching the animation, I stop it from time to time to reflect on what I am trying to get out of it? (N = 39; Missing = 8)	0 (0%)	1 (3.2%	1 (3.2%)	16 (51.6%)	13 (41.9%)
2.	Using Animated Examples help me to think logically during the animation process. (N = 39; Missing = 7)	0 (0%)	1 (3.1%)	1 (3.1%)	22 (68.6%)	8 (25.0%)
3.	I always discuss with my peers about the animation running process. (N = 39; Missing = 7)	0 (0%)	5 (15.6%)	9 (28.1%)	18 (56.3%)	0 (0%)

In week 10, the number of students who agreed/strongly agreed from the responses in questions 1 and 2, had increased significantly from 23 to 29 (77.8 to 93.5 percent) and

23 to 30 (74.2 to 93.6 percent), respectively. DIVTIC appeared to be more successful in supporting and encouraging students' higher-order thinking process in the long term. There was a significant increase in the numbers of students who reported in questions 1 (from 77.8 to 93.5 percent) and 2 (from 74.2 to 93.6 percent) that the use of DIVTIC could help them reflect their ideas and think logically throughout the animation process. However, the number of the students who agreed/strongly agreed in question 3 decreased from 20 to 18 (66.6 to 56.3 percent). This shows that the discussion between the students was less as the course progressed. A possible explanation may have been that as the course progressed, the weekly tasks were getting more complicated and more time was needed to concentrate on the animation process. Thus, the students did not have time to discuss anything with their peers. However, overall the results reveal that majority of the students did engage in a higher-order thinking process from time to time.

- **Confidence**

The following three questions sought to explore the way in which using DIVTIC increased the students' confidence. Table 6.28 shows the questions and the students' responses.

Table 6.28: Confidence (week 6)

	Confidence Week 6	SD	D	NA	A	SA
1.	Using Animated Examples increase my confidence in learning programming. (N = 39; Missing = 8)	1 (3.2%)	4 (12.9%)	7 (22.6%)	14 (45.2%)	5 (16.1%)
2.	Using Animated Examples, I believe that I can solve more complicated tasks. (N = 39; Missing = 8)	1 (3.2%)	6 (19.4%)	9 (29.0%)	14 (45.2%)	1 (3.2%)
3.	Using Animated Examples, I feel that I can help other peers in solving a given problem. (N = 39; Missing = 8)	0 (0%)	12 (38.7%)	9 (29.0%)	9 (29.0%)	1 (3.2%)

In week 6, question 1 explored whether using the animated examples component had increased students' confidence. Table 6.28 shows that 19 students (61.3 percent) agreed/strongly agreed that they experienced more confidence in programming after using the animated examples component, while there were only five students (16.1 percent) who disagreed/strongly disagreed. Question 2 sought to examine whether the students believed they were able to solve more complicated tasks or not. It shows that 15 students (48.4 percent) believed that they were able to solve more complicated tasks while seven students (22.6 percent) were not. Responses to question 3 show that the number of the students who agreed/strongly agreed and disagreed/strongly disagreed that they could help other peers in solving a given task were not significantly different.

Table 6.28a: Confidence (week 10)

	Confidence Week 10	SD	D	NA	A	SA
1.	Using Animated Examples increase my confidence in learning programming. (N = 39; Missing = 7)	1 (3.1%)	1 (3.1%)	5 (15.6%)	19 (59.4%)	6 (18.8%)
2.	Using Animated Examples, I believe that I can solve more complicated tasks. (N = 39; Missing = 8)	1 (3.2%)	5 (16.1%)	17 (54.8%)	7 (22.6%)	1 (3.2%)
3.	Using Animated Examples, I feel that I can help other peers in solving a given problem. (N = 39; Missing = 7)	1 (3.1%)	4 (12.5%)	16 (50.0%)	10 (31.3%)	1 (3.1%)

In week 10, the number of the students who responded to question 1 had increased significantly from 19 (61.3 percent) to 25 (78.2 percent). This reveals that many students had more confidence in learning programming after having used the animated examples component for a longer period of time. However, there was a significant decrease in the number of the students who responded to question 2 from week 6 (48.4 percent) to week 10 (25.8 percent). On the other hand, the number of the students (32.2 percent) who agreed/strongly agreed to question 3 from week 6 was approximately the same as in week 10 (34.4 percent), but there was a significant increase in the number of the students who commented 'not applicable' from 9 (29 percent) to 16 (50 percent). These results suggested that the animated examples component did accomplish the task of increasing students' confidence in learning programming, but it did not help them to solve more complicated tasks or to enable them to help other peers solve a given problem.

6.4.2 Motivation

- **Encouragement**

Table 6.29: Encouragement (week 6)

	Encouragement Week 6	SD	D	NA	A	SA
1.	Using Animated Examples, I feel that I pay more attention in programming class. (N = 39; Missing = 8)	0 (0%)	6 (19.4%)	8 (25.8%)	15 (48.4%)	2 (6.5%)
2.	Using Animated Examples encourage me in programming more efficiently. (N = 39; Missing = 8)	1 (3.2%)	3 (9.7%)	9 (29.0%)	12 (38.7%)	6 (19.4%)
3.	Using Animated Examples, I feel that programming is not too difficult to learn. (N = 39; Missing = 9)	1 (3.3%)	2 (6.7%)	11 (36.7%)	14 (46.7%)	2 (6.7%)

In using the animated examples component, Table 6.29 shows that there were 17 students (54.9 percent) who agreed/strongly agreed that they had paid more attention in programming class while eight students (25.8 percent) marked 'not applicable' and six students (19.4 percent) who disagreed. There were 18 students (67.7 percent) who agreed that animated examples encouraged them to programme more efficiently while nine students (29 percent) said it was 'not applicable' and four students (12.9 percent)

disagreed/strongly disagreed. There were 16 students (53.4 percent) who agreed/strongly agreed that they felt the programming was not too difficult to learn while 11 students (36.7 percent) said it was 'not applicable' and three students (10 percent) disagreed/strongly disagreed.

Table 6.29a: Encouragement (week 10)

	Encouragement Week 10	SD	D	NA	A	SA
1.	Using Animated Examples, I feel that I pay more attention in programming class. (N = 39; Missing = 7)	0 (0%)	1 (3.1%)	6 (18.8%)	22 (68.8%)	3 (9.4%)
2.	Using Animated Examples encourage me in programming more efficiently. (N = 39; Missing = 7)	0 (0%)	2 (6.3%)	9 (28.1%)	17 (53.1%)	4 (12.5%)
3.	Using Animated Examples, I feel that programming is not too difficult to learn. (N = 39; Missing = 9)	1 (3.2%)	2 (6.5%)	14 (45.2%)	11 (35.5%)	3 (9.7%)

In Week 10, in using the animated examples component, there were 25 students (78.2 percent) who agreed/strongly agreed that they had paid more attention in programming class while six students (18.8 percent) marked 'not applicable' and only one student (3.1 percent) disagreed. There were 21 students (65.6 percent) who agreed that it encouraged them to programme more efficiently while nine students (28.1 percent) said it was 'not applicable' and two students (6.3 percent) who disagreed/strongly disagreed. There were 14 students (45.2 percent) who agreed/strongly agreed that they felt that programming was not too difficult to learn while 14 students (45.2 percent) said 'not applicable' and three students (9.7 percent) disagreed/strongly disagreed.

The findings show that there was a significant increase in the number of students who agreed/strongly agreed that they paid more attention in programming class, from 17 to 25 (54.9 to 78.2 percent). This seemed to suggest that the animated examples component could be used to encourage students to pay more attention in class. The percentage of students who agreed/strongly agreed that it encouraged them to programme more efficiently (58.1 to 65.5 percent) increased as the course progressed. This seemed to suggest that the animated examples component could also be used to help students create learning outcomes more efficiently. However, the animated examples component did not seem to make students believe that programming was easier as the course progressed and the concepts and tasks became more complex.

- **User friendliness**

Table 6.30: User Friendliness (week 6)

	User Friendliness Week 6	SD	D	NA	A	SA
1.	The interface of Animated Examples is pleasant. (N = 39; Missing = 9)	0 (0%)	3 (10.0%)	6 (20.0%)	17 (56.7%)	4 (13.3%)
2.	Animated Examples are an easy-to-use tool. (N = 39; Missing = 8)	1 (3.2%)	0 (0%)	4 (12.9%)	19 (61.3%)	7 (22.6%)
3.	Animated Examples are easy to navigate. (N = 39; Missing = 8)	0 (0%)	0 (0%)	7 (22.6%)	17 (54.8%)	7 (22.6%)

Table 6.30 shows that in week 6, there were 21 students (70 percent) who agreed/strongly agreed that the animated examples component had an attractive interface. There were 26 students (83.9 percent) who agreed/strongly agreed that it was an easy tool to use, whereas only one student found it was difficult to use. In term of the navigation system, there were 24 students (77.4 percent) who agreed/strongly agreed that DIVTIC was easy to navigate.

Table 6.30a: User Friendliness (week 10)

	User Friendliness Week 10	SD	D	NA	A	SA
1.	The interface of Animated Examples is pleasant. (N = 39; Missing = 7)	0 (0%)	1 (3.1%)	3 (9.4%)	22 (68.7%)	6 (18.8%)
2.	Animated Examples are an easy-to-use tool. (N = 39; Missing = 7)	0 (0%)	1 (3.1%)	4 (12.5%)	18 (56.3%)	9 (28.1%)
3.	Animated Examples are easy to navigate. (N = 39; Missing = 7)	0 (0%)	0 (0%)	5 (15.6%)	20 (62.5%)	7 (21.9%)

In week 10, Table 6.30a shows that there was only one student who disagreed that the interface of DIVTIC was good, three who marked not applicable, and 28 (87.5 percent) who agreed/strongly agreed. There were 27 students (84.4) who agreed/strongly agreed that the animated examples component was an easy-to-use tool and was easy to navigate.

The number of students who agreed/strongly agreed that the interface of the animated examples component was pleasant substantially increased from week 6 to week 10 (21 to 28 students or 70 to 87.5 percent). This suggests that once the students had used the animated examples component for a period of time, they felt more likely to agree that it was designed to be a user-friendly tool. There was no real difference in the number of students who reported that the animated examples component was an easy to navigate and use. This seems to suggest that most of the students appeared to be satisfied with the features of the animated examples component in that was designed to be easy to use and navigate from the first use.

- **Perceived enjoyment**

Table 6.31: Perceived enjoyment (week 6)

Perceived Enjoyment Week 6	SD	D	NA	A	SA
1. I enjoy using Animated Examples. (N = 39; Missing = 8)	1 (3.2%)	1 (3.2%)	11 (35.5%)	15 (48.4%)	3 (9.7%)
2. Animated Examples entertain me in learning programming. (N = 39; Missing = 8)	1 (3.2%)	4 (12.9%)	6 (19.4%)	17 (54.8%)	3 (9.7%)
3. I feel comfortable by using Animated Examples. (N = 39; Missing = 8)	1 (3.2%)	3 (9.7%)	6 (19.4%)	16 (51.6%)	5 (16.1%)

Table 6.31 shows that in week 6 there were 11 students (35.5 percent) who marked not applicable that they enjoyed using the animated examples component whereas 18 students (58.1 percent) who agreed/strongly agreed. There were 20 students (64.5 percent) who agreed/strongly agreed that the animated examples component entertained them in learning programming while five students (16.1 percent) who did not. The number of the students who agreed/strongly agreed that they felt comfortable in using the animated examples component was up to 21 (67.7 percent) whereas there were only four students who did not.

Table 6.31a: Perceived enjoyment (week 10)

Perceived Enjoyment Week 10	SD	D	NA	A	SA
1. I enjoy using Animated Examples. (N = 39; Missing = 7)	0 (0%)	3 (9.4%)	9 (28.1%)	18 (56.3%)	2 (6.3%)
2. Animated Examples entertain me in learning programming. (N = 39; Missing = 7)	0 (0%)	6 (18.8%)	7 (21.9%)	17 (53.1%)	2 (6.3%)
3. I feel comfortable by using Animated Examples. (N = 39; Missing = 7)	0 (0%)	1 (3.1%)	5 (15.8%)	21 (65.6%)	5 (15.8%)

In week 10, however, Table 6.31a shows that there was no substantial change from week 6 to week 10. The number of students who agreed/strongly agreed that they enjoyed using the animated examples component, increased by 2 (from 18 to 20 students, or from 58.1 to 62.6 percent), while the number of students who disagreed/strongly disagreed, also increased by 1 (from 2 to 3 or from 6.4 to 9.4 percent). There was also no overall change in the number of the students who agreed/strongly agreed that the animated examples component entertained them in learning programming, in which the number of the students who agreed/strongly agreed had decreased by 1 (from 20 to 19), and those who did not agree, had increased by 1 (from 5 to 6). However, there was a large change in the number of students who agreed/strongly agreed that they felt comfortable in using the animated examples component from 21 to 26 students (67.7 to 81.2 percent).

The findings appeared to suggest that the majority of students were enjoying using the animated examples as the course progressed. It also seemed to suggest that the animated examples component could be used to entertain and motivate students throughout their learning process. In addition, most students were more comfortable in using the animated examples component toward the end of the study. This could suggest that the animated examples component was likely to be an enjoyable and entertaining tool that would make the students feel more comfortable after they had become familiar with it.

- **Perceived educational value**

Table 6.32: Perceived educational value (week 6)

	Perceived Educational Value Week 6	SD	D	NA	A	SA
1.	I am pleased to have Animated Examples as assistance.(N = 39; Missing = 8)	3 (9.7%)	0 (0%)	3 (9.7%)	14 (45.2%)	11 (35.5%)
2.	Animated Examples are a useful component in learning how to program in C. (N = 39; Missing = 8)	0 (0%)	2 (6.5%)	3 (9.7%)	17 (54.8%)	9 (29.0%)
3.	Animated Examples material is challenging. (N = 39; Missing = 8)	0 (0%)	3 (9.7%)	7 (22.6%)	19 (61.3%)	2 (6.5%)

Table 6.32 shows that there were only three students (97 percent) who strongly disagreed that they were pleased to use the animated examples component as learning tool, while there were 25 students (80.7 percent) who agreed/strongly agreed. There were 26 students (83.8 percent) who felt that the animated examples component was a useful component in learning how to program in C while there were only two students (6.5 percent) who did not. The number of the students who agreed/strongly agreed that the material from the animated examples component was challenging, was up to 21 (67.8 percent) while those who did not agree, were only 3 (9.7 percent).

Table 6.32a: Perceived educational value (week 10)

	Perceived Educational Value Week 10	SD	D	NA	A	SA
1.	I am pleased to have Animated Examples as assistance.(N = 39; Missing = 7)	0 (9.7%)	0 (0%)	2 (6.3%)	19 (59.4%)	11 (34.4%)
2.	Animated Examples are a useful component in learning how to program in C. (N = 39; Missing = 7)	0 (0%)	0 (0%)	2 (6.3%)	20 (62.5%)	10 (31.3%)
3.	Animated Examples material is challenging. (N = 39; Missing = 7)	0 (0%)	1 (3.1%)	11 (34.4%)	16 (50.0%)	4 (12.5%)

In week 10, Table 6.32a shows that there were 30 students (93.8 percent) who agreed/strongly agreed that they were pleased to use the animated examples component as an assistance tool, which was a useful component in learning how to program in C, while there was no one who did not agree. There were 20 students (62.5 percent) who

agreed/strongly agreed that the material from the animated examples component was challenging while there was only one student (3.1 percent) who disagreed.

There was a large change in the number of the students who agreed/strongly agreed that they were pleased to use the animated examples component as an assistant tool, from 25 to 30 (80.7 to 93.8 percent) and from 26 to 30 (83.8 to 93.8 percent) of those who supported that the animated examples component was useful in learning how to program in C. The findings seem to suggest that most of the students were likely to use the animated examples component as an instructional tool. Most of the students also agreed that the animated examples component was a useful tool in helping them to learn how to program as the course progressed. It could be seen that as the students used the animated examples component over a long period (from weeks 6 to 10), they appeared to recognise that the animated examples component was a valuable tool which could help them to gain perceived educational value. However, the challenge in the use of this component seemed to be established towards the end of the course.

- **Perceived level of interest**

Table 6.33: Perceived level of interest (week 6)

	Perceived Level of Interest Week 6	SD	D	NA	A	SA
1.	Using Animated Examples are boring. (N = 39; Missing = 8)	5 (16.1%)	9 (29.0%)	14 (45.2%)	2 (6.5%)	1 (3.2%)
2.	Animated Examples cause mental weariness. (N = 39; Missing = 8)	6 (19.4%)	12 (38.7%)	10 (32.3%)	1 (3.2%)	2 (6.5%)
3.	I falll asleep when I use Animated Examples. (N = 39; Missing = 8)	10 (32.3%)	12 (38.7%)	5 (12.8%)	3 (9.7%)	1 (3.2%)

Table 6.33 demonstrates that there were 14 students (45.1 percent) who disagreed/strongly disagreed and who marked not applicable to the statement that the animated examples component was boring, while there were three students (9.7 percent) who agreed/strongly agreed. There were 18 students who disagreed/strongly disagreed that the animated examples component caused mental weariness while 10 students (32.3 percent) who marked not applicable and only three students who agreed/strongly agreed. The number of the students who disagreed/strongly disagreed that they fell asleep when they used the animated examples component went up to 22 (71 percent) while there were five students (12.8 percent) who marked not applicable and four students (12.9 percent) who agreed/strongly agreed.

Table 6.33a: Perceived level of interest (week 10)

Perceived Level of Interest Week 10	SD	D	NA	A	SA
1. Using Animated Examples are boring. (N = 39; Missing = 7)	6 (18.8%)	14 (43.8%)	9 (28.1%)	2 (6.3%)	1 (3.1%)
2. Animated Examples cause mental weariness. (N = 39; Missing = 7)	8 (25.0%)	11 (34.4%)	9 (28.1%)	4 (12.5%)	0 (0%)
3. I fall asleep when I use Animated Examples. (N = 39; Missing = 7)	7 (21.9%)	12 (37.5%)	11 (34.4%)	2 (6.3%)	0 (0%)

In week 10, Table 6.33a shows that there was a large change in the number of students who disagreed/strongly disagreed that using the animated examples component was boring, from 14 to 20 (45.1 to 62.6 percent), while there was no change in the number of the students who agreed/strongly agreed. There was a small change in the number of the students who disagreed/strongly disagreed that the animated examples component caused mental weariness, from 18 to 19 (58.1 to 59.4 percent). However, the number of students who marked not applicable that they fell asleep when they used the animated examples component had increased from 5 to 11 students (12.8 to 34.4 percent).

The findings seem to suggest that students developed more interest in using the animated examples component as the course progressed. A few students had responded that the use of the animated examples component caused mental weariness in both two tests in weeks 6 and 10. The number of students who marked not applicable to falling asleep when they used this component, had strongly increased from 5 to 11 (12.8 to 34.4 percent). This shows that the students seemed to have some difficulties in solving more complicated tasks via the use of the animated examples component which could lead them to become tired when using this component.

The following table (Table 6.34) shows the summary of the problems occurred during the tutors' observation which involved students' attitude. These had to do with Interface, Log in, Weekly Task, and Network.

Table 6.34: Summary of Tutor observation

Problem	Detail
Interface	▪ Could not find a Play button?
Log in	▪ Some students could not log in because type the ID in upper case. ▪ Typed in the wrong URL
Weekly Task	▪ Did not understand how the DIVTIC Weekly Task work? ▪ In OPERATORS, student did not understand what had been asked for in problem 2, question 1. ▪ What do they need to do? ▪ In FUNCTION, problem 2, Q4, change *j = 6;* to *i = 6;*
Network	▪ Network is very slow.
Other	▪ Can they input their own codes?

The interface and the log-in problems were not actually the problems that caused students' negative attitudes. Once they knew where the Play button was, the problem was solved. In addition, with the log in problem, once the students knew that the ID and password had to be in lower case, they then used the lower case alphabet to log in. However, the weekly task problem aroused attitudes that were both negative and positive. The students who understood the weekly task questions and who could solve it appeared to be more likely to enjoy using DIVTIC to check their answers. On the other hand, when the weekly task questions were too complicated and unclear to them, they could feel that DIVTIC was not worth the time investment needed. One inquiry from a user which led to a positive attitude suggested that it would be more appropriate and interesting if they could test their own input via the use of DIVTIC.

6.4.3 Technical Satisfactory

- **Useability**

Both tutors agreed that students were satisfied with all elements of the animated examples component, except for the English explanation in the Message Board. Tutor A pointed out that the animated examples component helped the students' learning process by visualising the program execution step-by-step along with the relevant text explanation of each step. *"The program is quite clear and helps in less complicated matters. For example, at the' printf' or 'scanf' functions, the program shows calculation step by step... it's easy and quick to access the program, also there is a Menu to choose from... The program has a good explanation and is very detailed"*, stated Tutor A. His argument suggested that students had no difficulty with understanding how each element worked. This notion was supported by Tutor B, *"It is certainly easy for students - only clicking buttons. They don't need to have background... Students enjoy using this function. Wherever they click, there will be an explanation telling them what's happening."* The following table (Table 6.35) shows the student feedback from the DIVTIC self-administered evaluation questionnaire form.

Table 6.35: Useability (week 6)

	Useability Week 6	SD	D	NA	A	SA
1.	I can apply the concept getting from Animated Examples to solve given problems. (N = 39; Missing = 8)	0 (0%)	2 (6.5%)	10 (32.3%)	15 (48.4%)	4 (12.9%)
2.	Using Animated Examples enhance my understanding. (N = 39; Missing = 8)	1 (3.2%)	1 (3.2%)	5 (16.1%)	17 (54.8%)	7 (22.6%)
3.	Animated Examples cover all I need to learn in programming appropriately. (N = 39; Missing = 8)	1 (3.2%)	3 (9.7%)	5 (16.1%)	18 (58.1%)	4 (12.9%)

Table 6.35 shows that there were 19 students (61.3 percent) who agreed/strongly agreed that they could apply the concept gained from using the animated examples component to solve given problems, while 10 students (32.3 percent) who marked not applicable and two students (6.5 percent) who did not agree. There were 24 students (77.4 percent) who agreed/strongly agreed that the animated examples component enhanced their understanding, while five students (16.1 percent) marked not applicable and two students (6.4 percent) who did not agree. Twenty-two students (71 percent) agreed/strongly agreed that the animated examples component covered all they needed to learn programming appropriately while five students (16.1 percent) marked not applicable and four students (12.9 percent) did not agree.

Table 6.35a: Useability (week 10)

	Useability Week 10	SD	D	NA	A	SA
1.	I can apply the concept getting from Animated Examples to solve given problems. (N = 39; Missing = 7)	0 (0%)	2 (6.3%)	6 (18.6%)	20 (62.5%)	4 (12.5%)
2.	Using Animated Examples enhance my understanding. (N = 39; Missing = 7)	0 (0%)	0 (0%)	3 (9.4%)	25 (78.1%)	4 (12.5%)
3.	Animated Examples cover all I need to learn in programming appropriately. (N = 39; Missing = 7)	0 (0%)	3 (9.4%)	5 (15.6%)	22 (68.8%)	2 (6.3%)

In week 10, Table 6.35a shows that the number of students who disagreed/strongly disagreed that they could apply the concepts gained from the animated examples component to solve given problems was still the same as those in week 6, while the number of the students who agreed/strongly agreed had increased from 19 to 24 (61.3 to 75 percent). The number of students who agreed/strongly agreed that using the animated examples component enhanced their understanding had also increased from 24 to 29 (77.4 to 90.6 percent), while there was no one who disagreed. There was no change in the number of the students who agreed/strongly agreed that the animated examples component covered all they needed from weeks 6 to 10. The results seem to suggest that this component could be used to help them better understand the programming process as the course progressed. Overall, most of the students seemed to be happy with the useability of this component.

From the student interviews, all interviewees (24 students) agreed that DIVTIC was useful and interesting. One of the interviewees stated, *"I wish you teach other years' students too: Student 572"* and another one stated, *"I think it would be beneficial for ones who have used this program. Ones who haven't used it will need to do a self-study:*

Student 683." In addition, some interviewees found that DIVTIC was an easy tool to use after they got used to it:

> *Student 059: At first, I thought it's a bit confusing as my English is not so good. But after a while, about after 10 minutes I started to get used to it and found that it's not that hard... It's quite clear and is better than looking in the textbook. I can now see the picture and understand better.*
>
> *Student 938: It would be good to have picture to help understand.*
>
> *Student 872: I wasn't sure how it worked on the first week. This week is a bit better. I browsed through many times and the more I browse, the more I understand.*
>
> *Student 863: It's good. When I'm stuck with programming, I can look at and try to understand it. This program shows us each step and each line that memories are being used. It also shows on screen how much memory has been used.*

All interviewees appeared to be satisfied that they could take part in this study. They recommended changing some features to make it more comprehensive such as the English explanations should be changed into Thai, the speed of the animation should be slower, an audio explanation should be included in DIVTIC etc.

The findings seem to suggest that the animated examples component useful in that it covered all aspects appropriately, it could help the students apply concepts to solve given problems and it also enhanced their understanding. Most of the students seemed to be satisfied with the useability of this component.

- **Clarity**

Table 6.36 Clarity (week 6)

	Clarity Week 6	SD	D	NA	A	SA
1.	Concepts are addressed and well explained in Animated Examples. (N = 39; Missing = 8)	1 (3.2%)	3 (9.7%)	11 (35.5%)	14 (45.2%)	2 (6.5%)
2.	Animated Examples clarify me in learning the programming environment. (N = 39; Missing = 8)	0 (0%)	2 (6.5%)	13 (41.9%)	15 (48.4%)	1 (3.2%)
3.	Using Animated Examples, I comprehend how the program executes. (N = 39; Missing = 8)	0 (0%)	4 (12.9%)	2 (6.5%)	17 (54.8%)	8 (25.8%)

Table 6.36 shows that there were 16 students (51.7 percent) who agreed/strongly agreed that the concepts were addressed and well explained in the animated examples component while 11 students (35.5 percent) who marked not applicable and four students (12.9 percent) who did not agree. There were also 16 students (51.6 percent) who agreed/strongly agreed that the animated examples component helped them in learning the programming environment, while 13 students (41.9 percent) marked not applicable and two students (6.5 percent) who disagreed. The majority of students (80.6

percent) agreed/strongly agreed that they comprehended how the program was executed, while two students (6.5 percent) marked not applicable and four students (12.9 percent) disagreed.

Table 6.36a: Clarity (week 10)

	Clarity Week 10	SD	D	NA	A	SA
1.	Concepts are addressed and well explained in Animated Examples. (N = 39; Missing = 7)	0 (0%)	2 (6.3%)	3 (9.4%)	24 (75.0%)	3 (9.4%)
2.	Animated Examples clarify me in learning the programming environment. (N = 39; Missing = 7)	0 (0%)	1 (3.1%)	2 (6.3%)	24 (75.0%)	5 (15.6%)
3.	Using Animated Examples, I comprehend how the program executes. (N = 39; Missing = 7)	0 (0%)	1 (3.1%)	2 (6.3%)	20 (62.5%)	9 (28.1%)

In week 10, Table 6.36a shows that the number of students who agreed/strongly agreed that the concepts were addressed and well explained in the animated examples component, had increased from 16 to 27 (51.7 to 84.4 percent). The number of students had also increased from 16 to 29 (51.6 to 90.6 percent) of those who agreed/strongly agreed that the animated examples component clarified learning the programming environment. The number of the students had also increased from 25 to 29 (80.6 to 90.6 percent) of those who agreed/strongly agreed that they comprehended how the program was executed.

The findings seem to indicate that the animated examples component delivered concepts that were well addressed and well explained and which helped the students to learn the programming environment and program execution. Most of the students seemed to be satisfied with the clarity of the animated examples component.

- **Collaboration**

Table 6.37 Collaboration (week 6)

	Collaboration Week 6	SD	D	NA	A	SA
1.	I always discuss programming problems with my peers. (N = 39; Missing = 8)	0 (0%)	3 (9.7%)	10 (32.3%)	18 (58.1%)	0 (0%)
2.	I always post the programming problems to the WebBoard. (N = 39; Missing = 8)	2 (6.5%)	3 (9.7%)	12 (38.7%)	13 (41.9%)	1 (3.2%)
3.	It is more effective to learn Animated Examples in a group rather than individually. (N = 39; Missing = 8)	0 (0%)	1 (3.2%)	3 (9.7%)	17 (54.8%)	10 (32.3%)

Table 6.37 shows that there were 18 students (58.1 percent) who agreed that they discussed programming problems with their friends, while 10 students (32.3 percent) who marked not applicable and three students (9.7 percent) who disagreed. There were 14 students (45.1 percent) who agreed/strongly agreed that they always posted the programming problems to the C WebBoard, while 12 students (38.7 percent) who

marked not applicable and five students (16.2 percent) who disagreed/strongly disagreed. Twenty-seven students (87.1 percent) agreed/strongly agreed that it was more effective to use the animated example component in a group rather than individually, while three students (9.7 percent) who marked not applicable and one student (3.2 percent) who disagreed.

Table 6.37a: Collaboration (week 10)

	Collaboration Week 10	SD	D	NA	A	SA
1.	I always discuss programming problems with my peers. (N = 39; Missing = 7)	0 (0%)	0 (0%)	14 (43.8%)	16 (50.0%)	2 (6.3%)
2.	I always post the programming problems to the WebBoard. (N = 39; Missing = 7)	4 (12.5%)	7 (21.9%)	14 (43.8%)	6 (18.8%)	1 (3.1%)
3.	It is more effective to learn Animated Examples in a group rather than individually. (N = 39; Missing = 7)	0 (0%)	0 (0%)	4 (12.5%)	17 (53.7%)	11 (34.4%)

In week 10, Table 6.37a shows that there were 18 students (56.3 percent) who agreed/strongly agreed that they discussed programming problems with their friends, while 14 students (43.8 percent) marked not applicable. There were seven students (21.9 percent) who agreed/strongly agreed that they always posted the programming problems to the C WebBoard, while 14 students (43.8 percent) marked not applicable and 11 students (34.4 percent) who disagreed/strongly disagreed. Twenty-eight students (87.4 percent) agreed/strongly agreed that it was more effective to use this component in a group rather than individually, while four students (12.5 percent) marked not applicable.

The results show that approximately, half of the students seemed to have held discussions on programming problems with their friends, while the others seemed to solve their programming problems individually. There were some students who agreed/strongly agreed that they had posted some programming problems to the WebBoard. Once again, this result was conflicting with the log-in record which indicated that there was no activity via the use of C WebBoard throughout the entire study. Therefore, the finding concluded that the C WebBoard was unused. Most of the students seemed more likely to use this component as a group but this result also conflicted with the first question, "*I always discuss programming problems with my peer.*" One possible explanation of this may have come from the fact that students appeared to be too shy to ask or discuss with their peers.

6.4.4 Conclusions

DIVTIC seemed to be useful in encouraging and supporting the majority of students to think along with the animation process while the discussion between students decreased as the course progressed. The results also show that majority of students gained more confidence from using DIVTIC to help them solve any basic programming problems as the course progressed. However, DIVTIC seemed to be unable to help students solve more complicated tasks or enable them to help their peers as the course progressed.

The animated examples component could be used to encourage students to pay more attention in class and to create more efficient learning outcomes. Most students appeared to be satisfied with the features of the animated examples component that proved to be a user-friendly tool designed that was easy to use and navigate. They gained more enjoyment and felt more comfortable in using this component as the course progressed. The animated examples component seemed to be a useful component which covered all aspects appropriately that could help the students apply concepts to solve given problems and enhance their understanding. This component provided concepts that were well addressed and well explained and helped the students to learn the programming environment and program execution. Most of the students seemed more likely to use this component as a group but they were too shy to discuss or ask any questions.

CHAPTER 7: RESEARCH QUESTION 2

This chapter discusses research question 2, *To what extent does the dynamic interactive visualisation process implemented in DIVTIC influence learning outcomes?* The intention of this question was to explore how the use of this tool influenced students' achievement in this study so that findings associated with learning outcomes could be understood in relation to the use of this visualisation tool. Table 7.1 shows the question, method, data collection, and analysis conducted in answering this research question.

Table 7.1: Data matrix for research question 2

Research Question	Method	Data Collection	Data Analysis
2. To what extent does the dynamic interactive visualisation process implemented in DIVTIC influence learning outcomes?			
(2a) How does the dynamic interactive visualisation process implemented in DIVTIC influence students' performance in programming?	(2a) F	(2a, 2b) Collect lab test 1, midterm, lab test 2, and final scores from F (weeks 7, 7, 11, and 13, respectively)	(2a) Mean comparisons to investigate differences between the experimental and control groups
(2b) How does use of the dynamic interactive visualisation process implemented in DIVTIC vary among students?	(2b) F		(2b) Mean comparisons to investigate differences between each level of students' GPA in the experimental groups
(2c) What levels and forms of cognitive engagement are evident among DIVTIC users?	(2c) C, G, H, and I	(2c) Collect data from C, G, H, and I (weekly)	(2c) Descriptive analysis
(2d) What factors influence students' achievement with DIVTIC?	(2d) A and I	(2d) Collect data from A (weeks 6 and 12) and I (weekly)	(2d) Inferential analysis to investigate relationships between achievement and time spent in DIVTIC, and computer experiences.

NOTE:
A: DIVTIC self-administered evaluation questionnaires
B: Subject semi-structured interviews
C: Tutors' observations
D: Researcher's observations
E: Tutor semi-structured interviews
F: Lab tests 1 and 2, Midterm and final examinations
G: Screen recording
H: DIVTIC weekly task
I: DIVTIC log file

7.1 Question 2a: How does the dynamic interactive visualisation process implemented in DIVTIC influence students' performance in programming?

The methods used to collect data to answer this question included laboratory test 1, midterm examination, laboratory test 2, and the final examination which were taken in weeks 7, 7, 11, and 13, respectively. There were 17 students in the control group and 11 students in the experimental group who withdrew from the course. Thus, there were

only 72 students (Control Group = 33; Experimental Group = 39) who participated in this study. The following section discusses the performance of students in the control and experimental groups by comparing their achievement in all the tests between the two groups, experimental and control.

7.1.1 Students' Performance

The following table (Table 7.2) shows the mean and standard deviation in each of the 4 tests taken by students in the two groups.

Table 7.2: The mean and standard deviation achieved in each test by the students in the experimental and control groups

Group	No.	Lab Test 1		Midterm		Lab Test 2		Final	
		M	SD	M	SD	M	SD	M	SD
Experimental	39	5.83	2.72	15.92	2.83	5.36	2.70	29.04*	5.95
Control	33	5.51	2.30	17.01	3.49	5.92	2.42	31.74*	4.83

* $p < 0.05$

Table 7.2 shows the mean in the laboratory test 1 of the experimental group (m = 5.83) was slightly higher than the control group (m = 5.51). On the other hand, the means in other tests, midterm examination (m = 17.01), laboratory test 2 (m = 5.92), and the final examination (m = 31.74), of the control group were higher than those of the experimental group, (m = 15.92), (m = 5.36), and (m = 29.04) respectively.

A possible cause of these results is that the use of DIVTIC helped some students in the early stage of the course as evidenced by the laboratory test 1 but did not have an overall influence on all students. The students who did not use DIVTIC scored better in other tests and in particular scored significantly higher results in the final examination. There seem to be a number of possible explanations for this finding. For example, use of DIVTIC is time consuming and could be an obstruction for some students in learning programming since it takes away from time spent in solving actual problems and tasks given by the instructor. The two-hour laboratory session might not have been adequate for the students to complete a weekly task and the DIVTIC examples given by the researcher and the weekly problems given by the teacher. The results from the weekly tasks also show that there were approximately 30 percent of students who completed all the tasks, while 25 percent of students finished none. Another 45 percent of students completed some of the tasks.

7.1.2 Conclusion

The results suggest that DIVTIC may not be the solution for all students. The findings suggest this because the results from the whole group do not show significant achievement gains. While it was originally thought that use of DIVTIC might benefit all students, the results do not support this contention. Further investigations were conducted with the data to explore if DIVTIC assisted the learning of particular students.

7.2 Question 2b: How does use of the dynamic interactive visualisation process implemented in DIVTIC vary among students?

An inquiry was undertaken to explore how different students used DIVTIC. This was to explore the variations between students in learning computer programming by using DIVTIC as a tool supporting their learning. The methods used to collect data to answer this question included laboratory test 1, midterm examination, laboratory test 2, and the final examination.

In order to explore the impact of the use of DIVTIC on students' achievement, the data was organised to reveal levels of student achievement among students of different ability levels. The GPA of students was used to separate the students into three distinct groups to assist in this inquiry.

Table 7.3 shows the number of students who participated or withdrew from the study in each group and level. It shows that the majority of the students who withdrew from the course were those with a low GPA in both groups (C1 = 56.25 percent, E1 = 56.25 percent). However, there were no students with a high GPA in the experimental group who withdrew from the course. The extent of the withdrawal from among the low GPA strengthens arguments for use of such visualisation products as DIVTIC. Clearly, such students find programming difficult and would appear to benefit from the extra support and assistance.

Table 7.3: The number of participants and withdrawals in each group and level

Students	GPA 1.11 – 1.72		GPA 1.78 – 2.22		GPA 2.25 – 3.36	
	C1	E1	C2	E2	C3	E3
Participated Students	7 (43.75%)	7 (43.75%)	14 (82.35%)	15 (88.24%)	12 (70.59%)	17 (100%)
Withdrawal Students	9 (56.25%)	9 (56.25%)	3 (17.65%)	2 (11.76%)	5 (29.41%)	0 (0%)

The scores for each student in both groups including the laboratory test 1 (10 percent), midterm examination (30 percent), laboratory test 2 (10 percent), and final examination (50 percent) were entered into an Excel spreadsheet for analysis with SPSS. The variation between students was examined by analysing the influence of GPA in learners' achievement.

7.2.1 Influence of GPA

The following sections describe students' achievement based on their learning abilities, low GPA (1.11 – 1.72), average GPA (1.78 – 2. 22), and high GPA (2.25 – 3.36), by comparing the mean of each group in each test including laboratory test 1, midterm examination, laboratory test 2, and the final examination. Both laboratory tests 1 and 2 were considered to be minimal tests and were comprised of two small problems requiring the development of C source code to solve the problems. Each test was worth 10 percent of the overall mark for the trimester. On the other hand, the midterm and final examinations were both formal tests and worth 30 and 50 percent each, respectively. Each examination was comprised of 60 multiple-choice questions.

- **Students with a low GPA**

Table 7.4 shows the means and standard deviations of each test of the two groups in level 1 (Low GPA: 1.11 – 1.72).

Table 7.4: The mean and standard deviation of each test between the experimental and control groups in level 1

Group	N	Lab Test 1		Midterm Test		Lab Test 2		Final Test		Overall	
		M	SD	M	SD	M	SD	M	SD	M	SD
Experimental 1 (E1)	7	7.36	2.16	16.71*	3.43	6.36	2.59	32.03*	3.69	62.45	10.87*
Control 1 (C1)	7	5.07	2.11	13.57*	1.59	4.71	1.87	25.95*	2.82	49.31	5.77*

* $p < 0.05$

Table 7.4 reveals that students in Group E1, who used DIVTIC, had a higher mean in each test than those in Group C1. The t-test showed that the difference between the means of the two groups in laboratory test 1 (2.29) and laboratory test 2 (1.65) was not large enough to reach statistical significance. However, there were obviously relatively large differences between means in the midterm examination (3.14) with $t(8.48) = 2.20$, $p < 0.05$, and in the final examination (6.08) with $t(12) = 2.82$, $p < 0.05$. It appeared that the students may not have been using DIVTIC long enough by the laboratory test 1 for

the tool to make a difference. However, continued use throughout the trimester was found to contribute significantly to students' programming performance.

- **Students with an average GPA**

Table 7.5 shows the means and standard deviations of each test of the two groups in level 2 (Average GPA: 1.78 – 2.22).

Table 7.5: The mean and standard deviation of each test between the experimental and control groups in level 2

Group	N	Lab Test 1		Midterm Test		Lab Test 2		Final Test		Overall	
		M	SD	M	SD	M	SD	M	SD	M	SD
Experimental 2 (E2)	15	5.47	2.73	15.40	2.54	5.40	1.89	28.67*	5.94	54.93	8.35
Control 2 (C2)	14	5.43	2.53	17.36	3.04	5.75	2.60	32.98*	3.08	61.51	9.00

* $p < 0.05$

Students in Group C2 had a slightly higher mean than Group E2 in laboratory test 1 (0.04), in the midterm examination (1.96), and in the laboratory test 2 (0.35). However, students in Group C2 had a significantly higher mean in the final examination (4.31). The use of DIVTIC appeared to help the students in the early stage of the course, as evidenced by the laboratory test 1, but did not have the expected influence overall. Students who did not use DIVTIC scored significantly better. This was an unexpected finding in some ways but understandable in the light of DIVTIC and how it was used by students. The power of the tool was intended to lie in its ability and capacity to assist learners to develop strong mental models of the processing occurring in the conduct of various algorithms. The use of DIVTIC required a substantial time commitment, and this commitment generally limited the amounts of time students were able to spend attending to other tasks and programming activities. Clearly, students with an average GPA did not receive the level of learning benefit from the use of DIVTIC that those in the control group received from their alternative activities.

- **Students with a high GPA**

Table 7.6 shows the means and standard deviations of each test of the two groups in level 3 (High GPA: 2.25 – 3.36).

Table 7.6: The mean and standard deviation of each test between the experimental and control groups in level 3

Group	N	Lab Test 1		Midterm Test		Lab Test 2		Final Test		Overall	
		M	SD	M	SD	M	SD	M	SD	M	SD
Experimental 3 (E3)	17	5.53	2.70	16.06*	2.89	4.91	3.34	28.14*	6.57	54.64	13.02*
Control 3 (C3)	12	5.88	2.26	18.62*	3.56	6.83	2.31	33.68*	4.99	65.01	11.26*

* $p < 0.05$

The table reveals that students in Group E3 had lower mean scores than Group C3 in all tests. Students in Group C3 perform significantly better in all the tests. These results were very similar to those achieved by students with an average GPA. The results suggest that the use of DIVTIC did not appear to help students with a high GPA. The way the study used DIVTIC with high GPA students appeared detrimental to their learning. Students were required to use DIVTIC for 30 minutes in each laboratory session and for many high GPA students this appeared to be a non-productive time. They were not learning. It appeared that they already could visualise, already had good mental models and would have benefited from practice in programming more than visualisation tasks.

These represent important findings from the use of DIVTIC and from the use of any intervention used to enhance programming performance. The results indicate the need to target planned improvements to the groups of students with particular needs. Clearly, DIVTIC is a tool capable of enhancing learning among students with a low GPA. For students with an average or high GPA, mandatory use of this tool over existing methods was seen to actually impede performance and achievement of the students.

7.2.2 Learners' Interactions with DIVTIC

In order to further explore how the use of DIVTIC was seen to influence student achievement, the study explored the various forms of usage made by the students. In particular the study explored the ways learners interacted with the tool and sought to explore if the form of interaction played a significant role in their level of achievement.

- **Interaction with DIVTIC**

As discussed earlier in Section 6.2.2 that there were approximately 70 percent of the students who interacted with the use of DIVTIC by either pressing Stop, Play, Backward, or/and Forward buttons to control the animation process. Students who used the tool by making consistent use of these buttons to control the tool were categorised as having a high level of interactivity. Student who used the tool and who made limited use of these buttons were categorised as having a low level of interactivity.

The following table (Table 7.7) shows the mean and standard deviation of the overall score of the students who interacted with DIVTIC with low or high interaction in weeks 6 and 10.

Table 7.7: The mean and standard deviation of overall score between the students who had low or high interaction with DIVTIC

Level of Interaction	Week 6			Week 10		
	Number of Student	Overall score		Number of Student	Overall score	
		M	SD		M	SD
Low Interaction	9	53.09	7.83	3	49.78*	3.00
High Interaction	22	56.36	13.11	28	57.96*	10.98

* $p < 0.05$

The results show that there was no significant correlation coefficient between the mean and the overall score of the students in week 6. However, the students seemed to have more interactions towards the end of the course and the results show that there was a significant correlation between the mean and overall score in week 10 (8.19), with $t(10.28) = 3.03$, $p < 0.05$. A possible explanation of this may have come from the fact that thinking through the animation process and reflecting on outcomes through interactions with DIVTIC helped students to better understand the programming process.

The following table (Table 7.8) shows the number of students with a low, average, or high GPA who reported having a low or high level of interaction with DIVTIC in weeks 6 and 10.

Table 7.8: Number of students with a low, average, or high GPA who had low or high level of interaction with DIVTIC

Level of Interaction	Number of Student in Week 6 (n = 31)				Number of Student in Week 10 (n = 31)			
	Low GPA (n = 5)	Average GPA (n = 12)	High GPA (n = 14)	Total (n = 31)	Low GPA (n = 7)	Average GPA (n = 13)	High GPA (n = 11)	Total (n = 31)
Low Interaction	-	5 (41.67%)	4 (28.57%)	9 (29.03%)	-	3 (20.08%)	-	3 (9.68%)
High Interaction	5 (100%)	7 (58.33%)	10 (72.43%)	22 (70.97%)	7 (100%)	10 (79.92%)	11 (100%)	28 (90.32%)

Table 7.8 clearly shows that all students with a low GPA reported a high level of interaction in both weeks 6 and 10. This form of usage by the students with a low GPA may have been a factor contributing to their enhanced achievement. These students significantly outscored their counterparts in the control group in both the midterm and final examinations as shown in Table 7.4. Students with an average or high GPA appeared to have more interactions as the course progressed.

Another important factor that may have impacted students' achievement was collaboration among themselves. This study also explored the way learners collaborated with their peers and sought to explore if this form of collaboration played a significant role in level of achievement.

- **Discussion with peers while using DIVTIC**

Once again, as discussed earlier in Section 6.2.1 that there were approximately 51.6 percent of the students in week 6 who reported stopping the animation and discussing aspects with their peers, while the rest of them did not. However, in week 10 the percentage of the students who reported stopping the animation and discussing with their peers had increased a little from 51.6 to 54.3 percent. The following table (Table 7.9) shows the mean and standard deviation of the overall score of the students who stopped and discussed with their peers when they used DIVTIC, and those who did not, in weeks 6 and 10.

Table 7.9: The mean and standard deviation of overall score between the students who discussed with peers

Student	Week 6			Week 10		
	N (31)	Overall score		N (31)	Overall score	
		M	SD		M	SD
No Discussion	15	58.71	11.24	14	54.89	11.40
Discussion	16	52.31	11.79	17	59.05	10.10

* $p < 0.05$

The results show the number of the students who engaged in discussion with their peers tended to increase a little as the course progressed. However, there was no significant difference between the means of overall score in weeks 6 and 10. A possible reason may have come from the fact that students may have found the material covering more complexities in concepts requiring them to concentrate more on their own learning process within the limited allowed time. DIVTIC did not seem to encourage discussion among learners as verified by the log-in time showing that no one visited the WebBoard component during the study as shown in Figure 7.1.

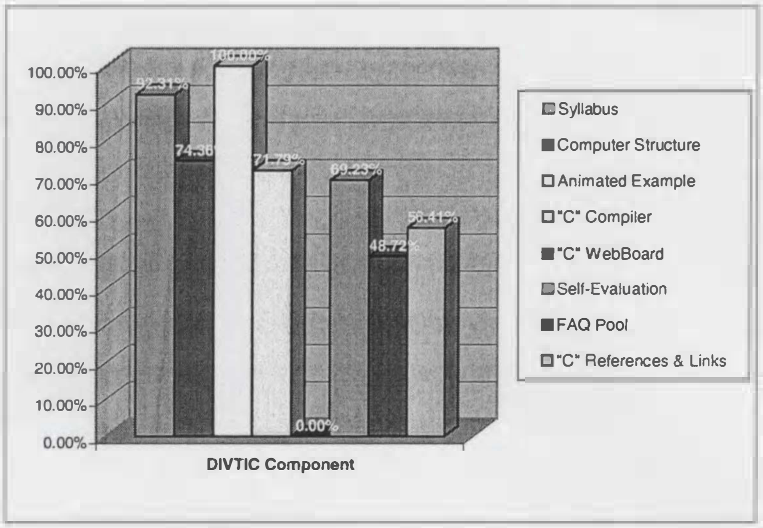


Figure 7.1: The usage of each component of DIVTIC from log files

7.2.3 Conclusion

The results from a quantitative study appeared to support the notion that use of DIVTIC can assist novices in learning introductory computer programming. The results are interesting in that they clearly demonstrate the advantage of DIVTIC with students with a low GPA. The students from this level in the experimental group, significantly outscored their counterparts in the control group in the final test suggesting that DIVTIC was an important element in their learning outcomes. However, the use of DIVTIC seemed to be of minimal value among students with an average or high GPA. The findings also suggest that students seemed likely to discuss with their peers consistently in class rather than using the C WebBoard. They also seemed likely to have more interaction with DIVTIC as the course progressed.

7.3 Question 2c: What levels and forms of cognitive engagement are evident among DIVTIC users?

An important contribution to learning is cognitive engagement. Students learn when they are actively engaged in a process that involves higher order thinking. The study sought to explore the ways in which DIVTIC encouraged and supported learners' engagement with the activities and examples.

The methods used to collect data in answering this question were tutor observations, screen recordings, DIVTIC weekly tasks, and DIVTIC log files. The use of each method is described as follows:

- **Tutor observation:** Tutors observed the laboratory session from weeks 2 to 12 except weeks 7 and 11, which was when the laboratory tests 1 and 2 were taken. The Tutor observation form was comprised of three open-ended questions as follows:
    - What questions did students ask when using DIVTIC?
    - What help did you need to give to students?
    - What problems did you face in this week relating to DIVTIC (Hardware, Software, Network, and Other)?
- **DIVTIC weekly task:** Each weekly task was comprised of two problems. Students were given the weekly task at the beginning of the laboratory and asked to do six consecutive steps:
    1. complete the weekly task;
    2. log into the DIVTIC system;
    3. run a relevant animation taken from the weekly task;
    4. write down the answer from DIVTIC;
    5. compare the answer to DIVTIC and write down a note if the answer was incorrect and why?; and,
    6. hand it back to tutors.

  There was no weekly task given in the first week since it was necessary to use this session to provide an introduction and preparation to the study. There was also no weekly task given in weeks 7 and 11, during which time the laboratory test 1 and 2, were given. Thus, there were nine weekly tasks, weekly tasks 1 to 9, in total. Each weekly task was marked and given a result in two categories including (a) score representing level of success in a programming task and (b) the level of success in the task requiring use of DIVTIC. A spreadsheet in Excel was used to calculate the frequency of each weekly task in order to explore the process which appeared to influence students' higher-order thinking, confidence, or motivation.

- **DIVTIC log files:** The log-in files were recorded and updated onto the database on a weekly basis starting from weeks 2 through to 11. At the end of the study, all the log-in files were put together into one big file and separated, using a perl script, into several files by using the students' ID as a name for each file.

7.3.1 Levels of cognitive engagement

In order to explore students' levels of cognitive engagement with DIVTIC, a screen video capture software was used to record the students' activities. Three students were recorded using DIVTIC for approximately 30 minutes weekly starting from weeks 3 to 12 and excluding weeks 7 and 11 in which laboratory tests 1 and 2 were given, respectively. This screen recording was used as a guideline to explore how students' use of DIVTIC might have influenced levels of cognitive engagement. The styles of students' use of DIVTIC were summarised into five levels of cognitive engagement based on their interactions with DIVTIC as shown in Table 7.10.

Table 7.10: Levels of cognitive engagement

Level	Action	Cognitive Engagement
1	Pressing Forward button to go to the end to see final result	Viewing initial final stage
2	Playing and watching	Observing variable and changes commands sequentially
3	Playing twice or more	Observing variable and changes commands sequentially, but repeating to review particular elements
4	Playing but stopping from time to time	Observing and predicting
5	Playing but stopping from time to time and repeating	Observing, predicting, and repeating to review particular elements

Table 7.10 shows 5 levels of cognitive engagement in which level 1 is used to represent a low level of cognitive engagement, while level 5 is used to represent a high level of cognitive engagement.

Table 7.11 shows numbers of students with different abilities associated with each level of cognitive engagement by combining levels 1 and 2 as a low level, level 3 as a medium level, and levels 4 and 5 as a high level. This data was observed from the screen recordings of 24 students.

Table 7.11: Levels of cognitive engagement of students with different GPA from screen recordings

Levels of Cognitive Engagement	Pattern	Number of Students			
		Low GPA (n = 5)	Average GPA (n = 7)	High GPA (n = 12)	Total (n = 24)
Low	Just pressing Forward button to go to the end of the animation to see the final results.	1	1	1	3 (12.50%)
	Playing and watching animations without interaction	2	1	2	5 (20.83%)
Medium	Playing and watching animations twice or more without interaction	-	1	2	3 (12.50%)
High	Playing and watching animations with interaction by pressing Stop, Pause, Step-Backward, Step-Forward, and Play buttons	1	2	3	6 (25.00%)
	Playing and watching animations twice or more with interaction by pressing Stop, Pause, Step-Backward, Step-Forward, and Play buttons	1	2	4	7 (29.17%)

Table 7.11 shows that there were only three students (12.5 percent) who did not watch the entire animation. They simply pressed the End button to go to the last frame of the animation to see the results. There were five students (20.83 percent) who watched the animation from the starting point through to the last frame by pressing the Play button at the beginning of the animation with no further interaction. There were also three students (12.5 percent) who just watched the animation two or three times from the starting point through to the last frame without interaction. However, there were six students (25 percent) who played and watched the animation with some interaction by pressing Play, Step-Forward, Step-Backward, Pause, and Stop buttons. In addition, there were seven students (29.17 percent) who also played and watched the animation two or three times with interaction.

The results show that use of DIVTIC seemed to influence the majority of students (54.17 percent) into a strong levels of cognitive engagement. These students showed the interaction with the animation by pressing, for example, the Pause, Step-Backward, and Play buttons to think through the process. Some students watched the animation frame-by-frame by pressing Step-Forward button at some specific point to review each concept. Some students even watched the animation twice or more with interaction to make sure that they understood the programming concepts. This action seemed to stimulate, extend, and enhance students thinking. Overall, the students with a high GPA seemed to have most interactions, while students with a low GPA had least interaction.

The following figure (Figure 7.2) shows the percentage of cognitive engagement for all 24 students.

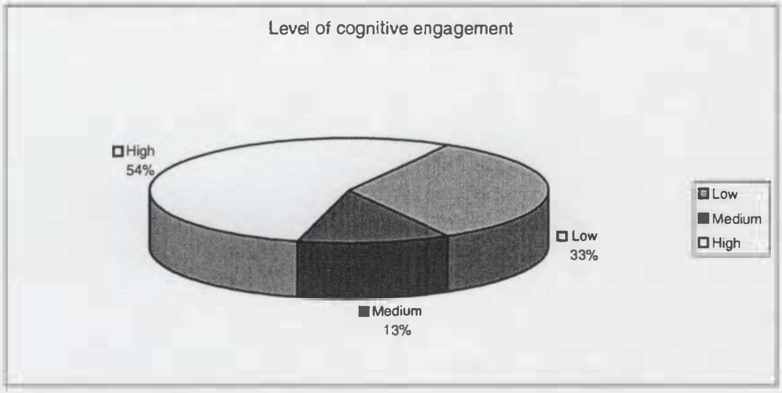


Figure 7.2: Percentage of the students with a low, medium, or high level of cognitive engagement

Figure 7.2 shows that there were approximately one third of the students (33 percent) who experienced low levels of cognitive engagement. The highest percentage of students, comparing the number of responses to the total number of each group, was the students with a low GPA (60 percent). On the other hand, the highest percentage of the students who had a high level of cognitive engagement was among students with an average (57.17 percent) or high GPA (53.85 percent). This finding may be explained in terms of the students' learning ability. Students with a low GPA might have made slow progress in learning which could have led them to interact less with DIVTIC since they had less time to play around with each component of the animation. It appeared that low GPA students simply watched the animation with less interaction. On the other hand, the average or high GPA students seemed likely to make more rapid progress with more interactions with DIVTIC and pressing buttons to go back and forth. These results suggest that for some students, use of DIVTIC did not require a high level of interactivity to support learning.

7.3.2 Level of Completion of Weekly Tasks

Students in the experimental group were provided with a weekly task sheet requiring them to predict outcomes and to use DIVTIC to check their answers. This process was designed to provide structure to the use of DIVTIC and to encourage students' use in a meaningful and deliberate fashion. The following sections discuss the level of completion of the weekly tasks among students and explore the extent to which this activity may have enhanced their learning.

Each weekly task was comprised of two small problems; each problem was repeated and was laid out into two columns; the left column contained a problem requiring students to write their own answers, while the right column contained the duplicate problem requiring students to write the answer obtained by watching the relevant animation from the DIVTIC system. Students first needed to use their knowledge to solve the problems in the weekly task sheet, and then they were asked to use DIVTIC to check their answers. This process was intended to help students closely concentrate on the animation to verify their answers.

Table 7.12 shows the numbers of students with different abilities who completed the weekly task activity from weeks 1 to 9 by comparing the answers from DIVTIC to their answers. This data was taken from the weekly task sheets that students handed in after using DIVTIC weekly.

Table 7.12: Weekly task—Using DIVITIC to compare the answers

Weekly Task	Using DIVTIC to Compare the Answers						
	Students with a Low GPA		Students with an Average GPA		Students with a High GPA		Total (%)
	N	%	N	%	N	%	
Week 1 (Flowchart)	7 (n = 7)	100	14 (n = 14)	100	16 (n = 16)	100	100
Week 2 (Data Type & I/O)	5 (n = 7)	71.43	5 (n = 14)	35.71	11 (n = 16)	68.75	56.80
Week 3 (Operators)	7 (n = 7)	100	12 (n = 14)	85.71	10 (n = 16)	62.50	82.90
Week 4 (Control)	6 (n = 7)	85.71	10 (n = 14)	71.43	14 (n = 16)	87.50	83.30
Week 5 (Functions)	7 (n = 7)	100	10 (n = 14)	71.43	12 (n = 16)	75	82.14
Week 6 (Arrays)	7 (n = 7)	100	11 (n = 14)	78.57	12 (n = 16)	75	84.52
Week 7 (Pointers)	5 (n = 5)	100	7 (n = 14)	50	10 (n = 16)	62.50	70.83
Week 8 (Sorting & Searching)	5 (n = 6)	83.33	8 (n = 14)	57.14	11 (n = 16)	68.75	69.74
Week 9 (Structures)	7 (n = 7)	100	11 (n = 14)	78.57	10 (n = 16)	62.50	80.35
Average		***93.38***		***69.84***		***73.61***	***75.65***

Table 7.12 shows that all students used DIVTIC to obtain the answers and to compare their answers in weekly task 1. However, the students with a low GPA appeared to be the group which was the highest level of completion of the task with an average of 93.38 percent. All students with a low GPA had compared their answers with the use of DIVTIC for 6 out of 9 weekly tasks, while all students in other groups, average and high GPA, had used it only once in the first week. This pattern seems to show a sign of continued motivation of students with a low GPA. This finding suggests that the use of weekly tasks along with DIVTIC had the potential to motivate students with a low GPA. Among average and high GPA students about 70 percent of students completed

the tasks. The remaining students used DIVTIC but in less formal ways. This seems to indicate that the weekly tasks may have encouraged students to use DIVTIC.

The following figure (Figure 7.3) show a graph of the percentage of all students who used DIVTIC to compare their answers from weekly tasks 1 to 9.

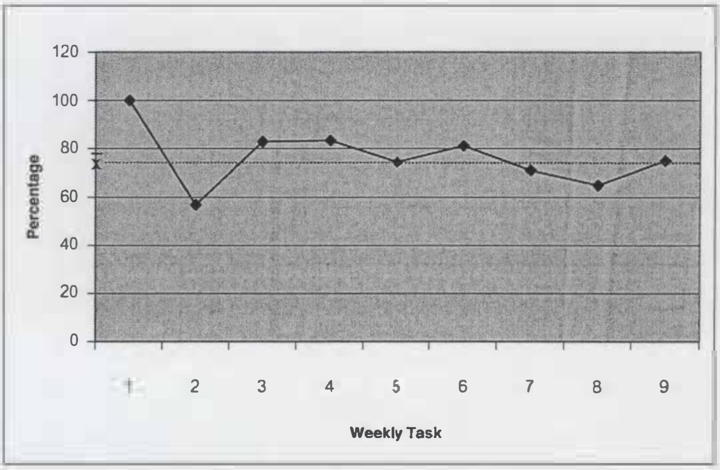


Figure 7.3: Students' use of DIVTIC for a comparison of the answers in weekly task

Figure 7.3 shows that the majority (75.65 percent) of students completed the weekly tasks as requested. The weekly task appeared to be a useful source that could help students engage in activities to assist their learning. The process of comparing the answers from the use of DIVTIC could stimulate, extend, and enhance students' thinking. In addition, students seemed to gain more confidence since they watched the animation and knew whether their answers were correct or not.

7.3.3 Level of Time Spent

In order to explore the level of time spent using DIVTIC, log-in time records from the log files were used to explore the ways in which students logged into the DIVTIC system and sought to explore if the log-in time had an impact to the level of cognitive engagement.

The records of log-in time were divided into two stages, first stage and second stage. The first stage was comprised of log-in time from weeks 1 to 5, while the second stage was comprised of log-in time from weeks 6 to 10. The following table (Table 7.13) shows the means in each stage of students' log-in time among the different ability groups.

Table 7.13: Two stages of log-in time of each level

Group	First Stage (Weeks 1 to 5)		Second Stage (Weeks 6 to 10)	
	Mean	Std. Deviation	Mean	Std. Deviation
Low GPA (n = 7)	394.86*	203.87	142.00*	134.99
Average GPA (n = 15)	265.73*	232.01	89.40*	99.00
High GPA (n = 17)	205.59*	132.85	75.94*	65.23

* $p < 0.05$

Table 7.13 shows that there was a significant decrease in the differences between the means of log-in time in both stages of the students with a low GPA ($t(6) = 4.519$, $p < 0.05$), students with an average GPA ($t(14) = 3.582$, $p < 0.05$), and students with a high GPA ($t(16) = 4.159$, $p < 0.05$). The mean of students with a low GPA ($m = 394.86$) in the first stage was higher than students with an average GPA ($m = 265.73$) and students with a high GPA ($m = 205.59$). This pattern was also repeated in the second stage. There was also an indication of a significant difference between the means of log-in time in the first stage of students with a low GPA and high GPA ($t(22) = 2.711$, $p < 0.05$). The results appeared to suggest that students with a lower GPA tended to spend more time using DIVTIC than others and as time progressed, all students made less use of it. Even with less time the students still seemed to get as much done.

7.3.4 Level of Students' Abilities in Answering the Weekly Tasks

Each student's outcome from the weekly task was divided into four different types of completion to explore how students did the problems. Results were recorded as follows: (a) no tasks completed, (b) tasks completed with many mistakes, (c) tasks completed with few mistakes, and (d) tasks completed with all correct as shown in Table 7.14.

Table 7.14: Weekly task—Percentage of completion

Weekly Task	No Tasks Completed		Tasks Completed with Many Mistakes		Tasks Completed with Few Mistakes		Tasks Completed with All Correct	
	N	Percent	N	Percent	N	Percent	N	Percent
1 (n = 37)	1	2.70	21	56.80	4	10.80	11	29.70
2 (n = 37)	9	24.30	21	56.80	5	13.50	2	5.40
3 (n = 35)	3	8.60	8	22.90	14	40.00	10	28.60
4 (n = 36)	4	11.10	3	8.30	22	61.10	7	19.40
5 (n = 37)	13	35.10	20	54.10	4	10.80	0	0
6 (n = 37)	9	24.30	11	29.70	8	21.60	9	24.30
7 (n = 31)	10	32.30	15	48.40	5	16.10	1	3.20
8 (n = 37)	15	40.50	20	54.10	2	5.40	0	0
9 (n = 32)	15	46.90	8	25.00	8	25.00	1	3.10

Table 7.14 shows that the majority of students successfully completed the problems in weekly task 1 (Flowchart). There was only one student (2.7 percent) who did not complete the problems in week 1. There were 21 students (56.8 percent) who completed the problems with many mistakes. The number of students who completed the problems with few mistakes was four (10.8 percent), while the number of students who completed the problems with all correct was 11 (29.7 percent).

In weekly task 2 (Data Types & I/O), the number of students who did not complete the problems had increased from one (2.7 percent), in weekly task 1, to nine (24.3 percent), while the number of students who completed with all correct had decreased from 11 (29.7 percent) to two (5.4 percent). The numbers of students who completed with many mistakes and with few mistakes in weekly tasks 1 and 2 were consistent. This weekly task was the beginning of the programming process involving reading source code and comprehension processes. Students seemed to have difficulty in coming to understand syntax and concepts even at the very beginning stages.

In weekly task 3 (Operators), the number of students who did not complete the problems had decreased from nine (24.3 percent), in weekly tasks 2, to three (8.6 percent) and those who completed with many mistakes had also decreased from 21 (56.8 percent) to eight (22.9 percent). On the other hand, the number of students who completed with few mistakes went up from five (13.5 percent) to 14 (4 percent) and those who completed with all correct also went up from two (5.4 percent) to 10 (28.6 percent). These numbers suggested a successful pattern was developing. The results suggest that students had tried to push themselves will more effort to complete the problems with all correct. This seemed to suggest that students had more confidence and motivation in solving programming problems from previous weeks. The number of students appeared to be consistent in solving problems in weekly task 4 (Control).

However, there was a sign of an unsuccessful pattern in weekly task 5 (Functions). The number of students who did not complete the problems went up from four (11.1 percent), in weekly tasks 4, to 13 (35.1 percent) and those who completed with many mistakes also went up from three (8.3 percent) to 20 (54.1 percent). On the other hand, the number of students who completed with few mistakes went down from 22 (61.1 percent) to four (10.8 percent) and those who completed with all correct also went down from seven (19.4 percent) to none. This weekly task was difficult and involved all

functions in the C language. Students seemed to have a difficult time coming to understand such a complicated concept. The results seemed to suggest that this topic (Functions) was very difficult for students to perceive and to move on from basic programming concepts such as Data Types, Input/Output, Operators, and Control to a more advance topic like Functions. The results also suggested that one-week duration to complete the *Functions* topic was inadequate. Students seemed to need more time for this big gap between the basic and advanced concepts. Students appeared to lose their motivation or confidence when faced with these complicated topics.

In weekly task 6 (Arrays), the number of students who did not complete the problems decreased from 13 (35.1 percent), in weekly tasks 5, to nine (24.3 percent) and those who completed with many mistakes had also decreased from 20 (54.1 percent) to 11 (29.7 percent). On the other hand, the number of students who completed with few mistakes went up from four (10.8 percent) to eight (21.6 percent) and those who completed with all correct also went up none to nine (24.3 percent). The results suggested that students seemed to have their motivation back from the previous weeks as was indicated in the number of students who completed with all correct.

The numbers of students who attended the class from weeks 1 to 6 were consistent. However, in weekly task 7 (Pointers), there was a significant decrease in the number of students who attended the class from 37, in weekly tasks 6, to 31. One of the possible reasons could be the difficulty of the previous topic, weekly task 6 (Functions), and as well as this weekly task 7 (Pointers). There were 10 students (32.3 percent) who did not complete the problems, 15 students (48.4 percent) who completed with many mistakes, and five students (16.1 percent) who completed with few mistakes. There was only one student (3.2 percent) who completed with all correct. The results appeared to suggest that some students had lost the necessary motivation when faced with this complicated topic. They seemed to be unwilling to attend the class afterwards. The number of students who completed with all correct had decreased from nine (24.30 percent) to one (3.20 percent).

In weekly task 8 (Sorting & Searching), the number of students who did not complete the problems increased from 10 (32.3 percent), in weekly tasks 7, to 15 (40.5 percent) and those who completed with many mistakes also increased from 15 (48.4 percent) to 20 (54.1 percent). On the other hand the number of students who completed with few

mistakes decreased from five (16.1 percent) to two (5.4 percent) and there was no one who completed with all correct. One of the possible reasons could be that this topic, *Sorting & Searching*, was also complicated and difficult to understand.

Once again, the number of students who attended the class was decreased from 37, in previous week, to 32 in weekly task 9 (Structures). The number of students who did not complete the problems was consistent, while the number of students who completed with many mistakes had decreased from 20 (54.1 percent), in weekly tasks 8, to eight (25 percent). The number of students who completed with few mistakes had increased from two (5.4 percent) to eight (25 percent) and who completed with all correct had increased from none to one (3.1 percent). The results appeared to suggest that this weekly task was easier to understand than the previous ones. The following figure (Figure 7.4) shows a graph of a relationship between weekly tasks 1 to 9 and the students' outcomes.

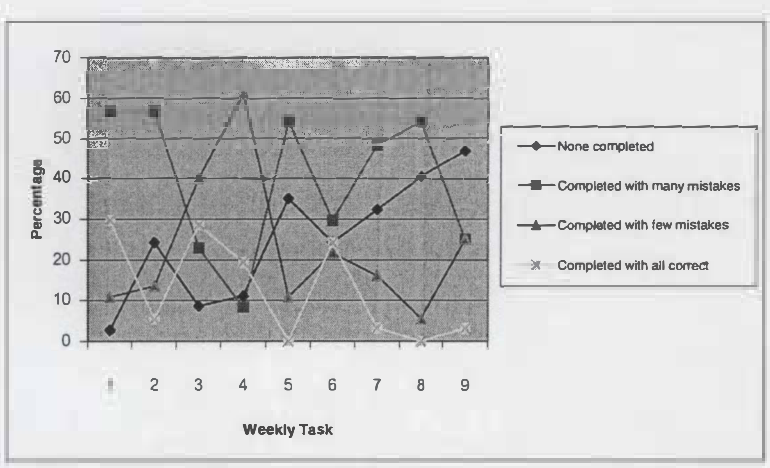


Figure 7.4: Students' outcomes from weekly tasks

The majority of the students had completed the problems with many mistakes, while the minority of the students had completed the problems without any mistakes. These results appeared to suggest that many of the weekly tasks were too difficult for students to complete with all correct, especially in weeks 5 and 8. There was much success experienced in early weeks. Predominantly most people did more tasks, or did some tasks with many of mistake. In some weeks, some students did very few tasks and had very little success with them. In some weeks, the tasks must have been too hard. The level of difficulty of tasks may need to be reviewed to ensure they provide the necessary assistance for students.

Figure 7.5 shows the average of each weekly task completed by each group of students. The completion of weekly tasks was divided into 4 levels:

- Level 1: no tasks completed;
- Level 2: tasks completed with many mistakes;
- Level 3: tasks completed with few mistakes; and
- Level 4: tasks completed with all correct.

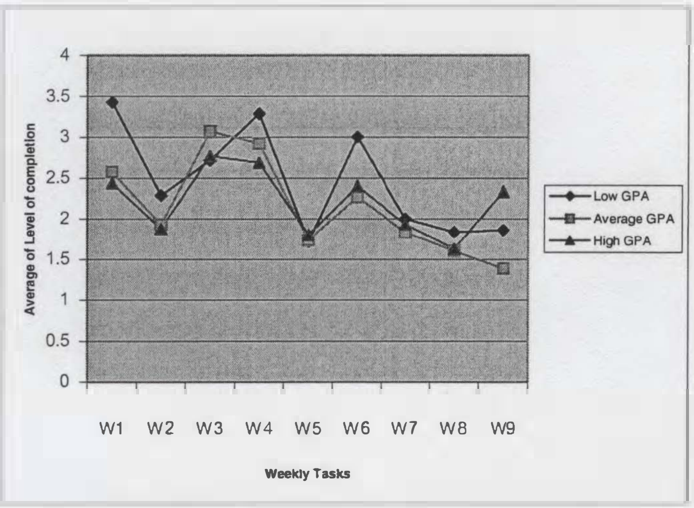


Figure 7.5: Average of weekly tasks completed by students with a low, average, and high GPA

The average level of weekly task completion of students with a low, average, and high GPA for the whole study were 2.46, 2.14, and 2.20, respectively. The results seemed to suggest that students with a low GPA appeared to have the highest level of weekly task completion. Thus, these students seemed to use DIVTIC the most. As discussed in Section 7.2.1, this finding seems to support and verify the achievement of students with a low GPA.

7.3.5 Conclusion

The results showed that students with an average or high GPA seemed likely to have more interaction with the use of DIVTIC as the course progressed than those with a low GPA. On the other hand, the students with a low GPA appeared to be the group with the highest level of weekly task completion, and used DIVTIC significantly more than other students. Their achievement significantly outscored their counterparts in the control group. This finding seems to support the notion that use of DIVTIC with less interaction

and completion of the task could assist novices in learning introductory computer programming successfully.

7.4 Question 2d: What factors influence students' achievement with DIVTIC?

In order to explore the impact of those factors influencing students' achievement, the data was organised to explore whether there were any obvious correlations between achievement and other variables. The methods used to collect data to answer this question included the DIVTIC self-administered evaluation questionnaires and the DIVTIC log files. The use of each method is described in more detail below:

- **The DIVTIC self-administered evaluation questionnaire form:** This form was used twice, in weeks 6 and 10. It contained three parts: (1) questionnaire, (2) checklist, and (3) open-ended question. The questionnaire was comprised of 11 scales using a five-point Likert rating scale which questioned the students' perceptions of the impact of DIVTIC on their higher-order thinking, confidence, motivation, user friendliness, enjoyment, interest, level of boredom, useability, clarity, collaboration, and experience. However, only the questionnaire which included experience patterns was used to answer this research question. The statistics software application, SPSS, was used to calculate frequency distributions.
- **DIVTIC log files:** The log-in files were recorded and updated onto the database on a weekly basis starting from weeks 2 through to 11. At the end of the study, all the log-in files were put together into one big file and separated, using a perl script, into several files by using the students' ID as a name for each file.

The statistics software application, SPSS, was used to calculate frequency distributions.

7.4.1 Time Factors

Many students appeared to spend different amounts of time using DIVTIC. These differences appeared to have impacted on their achievement. The log-in time was used to explore whether there was a significant correlation between time spent using DIVTIC

and students' achievement by using three different log-in times, weeks 2 to 6, weeks 7 to 11, and weeks 2 to 11, and the overall score. The following table (Table 7.15) is provided to explore the different between log-in time weeks 2 to 6, weeks 7 to 11, and weeks 2 to 11.

Table 7.15: Log-in time weeks 2 to 6, 7 to 11, and 2 to 11

Log-in Time	Low GPA: 1.11 - 1.72 (N = 7)		Average GPA: 1.78 – 2.22 (N = 15)		High GPA: 2.25 – 3.36 (N = 17)	
	M	SD	M	SD	M	SD
Weeks 2 to 6	394.86*	203.87	265.73	232.01	205.59*	132.85
Weeks 7 to 11	142.00	134.99	89.40	99.00	75.94	65.24
Weeks 2 to 11	536.86*	355.07	355.07	301.60	281.41*	165.32

* $p < 0.05$

Table 7.15 shows that there was a significant difference between the means of log-in time weeks 2 to 6 of the students with a low GPA and students with a high GPA ($t(22) = 2.711$, $p < 0.05$), and a significant difference between the means of log-in time weeks 2 to 11 of the students with a low GPA and students with a high GPA ($t(22) = 2.637$, $p < 0.05$). In fact, the means of log-in time weeks 2 to 6 (m = 394.86) and weeks 7 to 11 (m = 142.00) of the students with a low GPA were highest, while the means of log-in time weeks 2 to 6 (m = 205.59) and weeks 7 to 11 (m = 75.94) of students with a high GPA were the lowest. These results seemed to suggest that students with a low GPA were more likely to need DIVTIC as a tool in helping their learning process than others.

Figure 7.6 shows the weekly average of the log-in time between the first stage, weeks 2 to 6 and second stage, weeks 6 to 11.

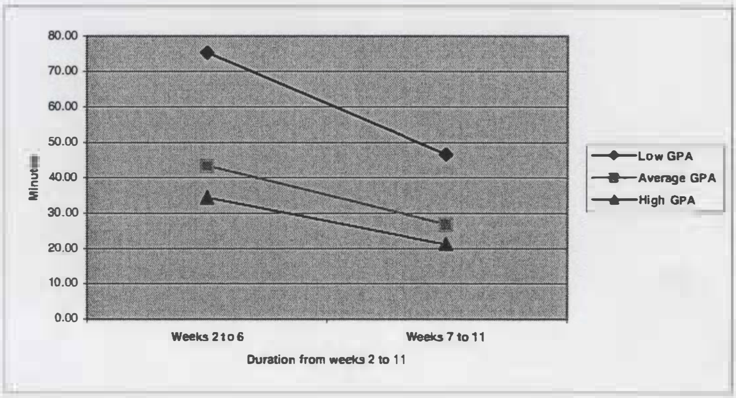


Figure 7.6: Average time spent using DIVTIC between weeks 2 to 6 and weeks 7 to 11

The results show that the means of log-in time decreased from the first half of the study, weeks 2 to 6, to the second half of the study, weeks 7 to 11. As time progressed students spent less time with DIVTIC and completed fewer tasks successfully. The harder tasks may have reduced students' motivation. Some possible explanations for the patterns of usage may have come from the fact that students with a low GPA made slow progress in learning which led them to use DIVTIC more than others, or that they might find out that DIVTIC was a useful tool to help them in learning how to program effectively. However, the results also suggest that all students might have difficulty in using DIVTIC toward the end of the course since there were more complexities in concepts as indicated by the decrease of the weekly average of log-in time from the first to second stages.

Clearly, students with a low GPA had a significant difference in their log-in times, weeks 2 to 6 and 2 to 11, over others in the experimental group as shown in Table 7.15. Further exploration into students' log-in times in each group and the scores of each test was carried out to see if there was any pattern in students' achievement in both groups.

Table 7.16 shows that students with different learning abilities in the experimental group, averaged log-in times from weeks 2 to 6, weeks 2 to 10, and the scores of each test.

Table 7.16: Levels of students in the experimental group with scores

Experimental Group	N	Log-in Weeks (minutes) 2 - 6 M	2 - 11 M	Lab Test 1 (10%) M	SD	Midterm Score (30%) M	SD	Lab Test 2 (10%) M	SD	Final Score (50%) M	SD
Low GPA	7	394.86*	536.86*	7.36	2.61	16.71	3.43	6.36	2.53	32.03	3.69
Average GPA	15	265.73	355.07	5.47	2.72	15.40	2.54	5.40	1.89	28.67	5.94
High GPA	17	205.59*	281.41*	5.53	2.70	16.06	2.89	4.91	3.34	28.14	6.57

* $p < 0.05$

Table 7.16 shows a pattern with students with a low GPA having means in all tests higher than those with an average or high GPA. The mean of each test for students with a low GPA, who used DIVTIC the most, seems to support the notion of their success and achievement being supported by more experience, resulting in enhanced learning outcomes. The findings seem to suggest that DIVTIC should be used for long periods of time in order to make the most difference to learning.

Table 7.17 shows the results of students with different learning abilities in the control group and the score of each test. The results appeared to show a pattern with students with a high GPA having the highest mean of each test over others and students with an average GPA having a higher mean in each test over students with a low GPA.

Table 7.17: Levels of students in the control group with scores

Control Group	N	Lab Test 1 (10%)		Midterm Score (30%)		Lab Test 2 (10%)		Final Score (50%)	
		M	SD	M	SD	M	SD	M	SD
Low GPA	7	5.07	2.11	13.57	1.59	4.71	1.87	25.95	2.82
Average GPA	14	5.43	2.53	17.36	3.04	5.75	2.60	32.98	3.08
High GPA	12	5.87	2.26	18.62	3.56	6.83	2.31	33.68	4.99

Table 7.18 shows a significant correlation between log-in time of all students in the first stage, weeks 2 to 6, and the laboratory test 1 ($r = 0.371$, $n = 39$, $p = < 0.05$, one-tailed). The results suggest that log-in time can influence students' achievement in the early stages of usage.

Table 7.18: Correlation between log-in time of all students from weeks 2 to 6 and the laboratory test 1

		Laboratory Test 1	All Students' Log-in Time Weeks 2 to 6
Laboratory Test 1	Pearson Correlation	1	.371*
	Sig. (1-tailed)	.	.010
	N	39	39
All Students' Log-in Time Weeks 2 to 6	Pearson Correlation	.371*	1
	Sig. (1-tailed)	.010	.
	N	39	39

* .Correlation is significant at the 0.05 level (1-tailed).

However, Table 7.19 shows that there was no significant correlation between log-in time of all students from weeks 2 to11 and the final examination ($r = -0.004$, $n = 39$, $p > 0.05$, one-tailed).

Table 7.19: Correlation between log-in time of all students from weeks 2 to 11 and the final examination

		Final Examination	All Students' Log-in Time Weeks 2 to 11
Final Examination	Pearson Correlation	1	-.004
	Sig. (1-tailed)	.	.490
	N	39	39
All Students' Log-in Time Weeks 2 to 11	Pearson Correlation	-.004	1
	Sig. (1-tailed)	.490	
	N	39	39

This result appeared to suggest that the overall log-in time did not influence students' achievement over the entire study. This could be because students were making so little use of DIVTIC. If they had been making more use, the outcome could have been different. Since, the students used DIVTIC in the second stage, weeks 7 to 11, obviously less than the early stage, weeks 2 to 6, a paired t-test was used to find out there was any significant difference between the means of log-in times. The result shows that there was a significant difference between the means of the two log-in times ($t(38) = 6.579$, $p < 0.0005$). This finding appears to explain why there was no significant correlation between log-in time from weeks 2 to11 and the final examination.

Table 7.20 shows that the correlation was significant between log-in time of students with a low GPA from weeks 2 to 11 and the final examination ($r = 0.683$, $n = 7$, $p = < 0.05$, one-tailed). Although, the number of students with a low GPA was small ($n = 7$), there was still a correlation.

Table 7.20: Correlation between log-in time from weeks 2 to 11 of students with a low GPA and the final examination

		Final Examination	Students with Low GPA' s Log-in Time Weeks 2 to 11
Final Examination	Pearson Correlation	1	.683*
	Sig. (1-tailed)	.	.045
	N	7	7
Students with Low GPA' s Log-In Time Weeks 2 to 11	Pearson Correlation	.683*	1
	Sig. (1-tailed)	.045	.
	N	7	7

* .Correlation is significant at the 0.05 level (1-tailed).

However, the log-in time of students with a low GPA in the first stage, weeks 2 to 6, showed no correlation to either the laboratory test 1 or midterm examination. The results appeared to verify that continued use of DIVTIC throughout the course would contribute to students' achievement.

Table 7.21 shows log-in times (weeks 2 to 6, 7 to 11, and 2 to 11) and the scores of each student with a low GPA. The results suggest that students with low GPA who used DIVTIC most were likely to achieve higher scores.

Table 7.21: Log-in time and scores of each student with low GPA

No.	Log-in Time (minutes)			Scores				
	Weeks 2 to 6	Weeks 7 to 11	Weeks 2 to 11	Lab Test 1 (10%)	Midterm (30%)	Lab Test 2 (10%)	Final (50%)	Total (100%)
1	245	20	265	2.5	15.5	2	26	46
2	182	85	267	9.5	13.5	5	32	60
3	277	142	419	6	13.5	7	29	56
4	281	157	438	10	22	9.5	35	76
5	516	25	541	9	19	8	34	70
6	513	145	658	6.5	14	5	32	57
7	750	420	1170	8	19.5	8	37	72

7.4.2 Computer Experience Factors

In this trimester, the students were using computer in a number of units. For many students, this trimester was their first serious computer-based learning experience. The results from the DIVTIC self-administered evaluation questionnaire form in part 1 was used to explore the experience students had towards the use of DIVTIC. This form was given to students in weeks 6 and 10. The statistics software application, SPSS, was used to calculate frequency distributions. The averages of each response were used to explore if there were any correlation between students' experience in using a computer and their outcomes.

Table 7.22: Experience (week 6)

Experience Week 6	SD	D	NA	A	SA
1. I have used a computer for many years. (N = 39; Missing = 8)	0 (0%)	4 (12.9%)	8 (25.8%)	9 (29.0%)	10 (32.3%)
2. I am an experienced Internet user. (N = 39; Missing = 8)	0 (0%)	6 (19.4%)	15 (48.4%)	7 (22.6%)	3 (9.7%)
3. I have no difficulty in using computer. (N = 39; Missing = 8)	0 (0%)	3 (9.7%)	13 (41.9%)	9 (29.0%)	6 (19.4%)

In week 6, Table 7.22 shows that there were 19 students (61.3 percent) who claimed that they had experience in using a computer for many years, but only four students (12.9 percent) did not have. There were 10 students (32.3 percent) who reported that they were experienced Internet users, while the majority of students, 15 students (48.4 percent) reported that they were not applicable and six students (19.4 percent) reported that they were not experienced Internet users. There were 15 students (48.4 percent) who reported having no difficulty in using a computer, while 13 students (41.9 percent) reported not applicable and three students reported having difficulty in using a computer.

Table 7.23: Experience (week 10)

Experience Week 10	SD	D	NA	A	SA
1. I have used a computer for many years. (N = 39; Missing = 7)	0 (0%)	4 (12.5%)	0 (0%)	16 (50.0%)	12 (37.5%)
2. I am an experienced Internet user. (N = 39; Missing = 7)	0 (0%)	4 (12.5%)	4 (12.5%)	18 (56.3%)	6 (18.8%)
3. I have no difficulty in using computer. (N = 39; Missing = 7)	2 (6.3%)	3 (9.4%)	6 (18.8%)	14 (43.8%)	7 (21.9%)

Table 7.23 shows that in week 10, there were 28 students (87.5 percent) who claimed that they had experience in using computer for many years, but only four students (12.9 percent) did not. There were 24 students (75.1 percent) who claimed that they were experienced Internet users, but four students (12.5 percent) norminated not applicable and other four students (12.5 percent) were not experienced Internet users. Twenty-one students (65.7 percent) claimed that they had no difficulty in using computer, six students (18.8 percent) were not applicable, and five students (15.7 percent) had difficulty.

The results show that by week 10, students' perception of the computer experience had changed. They gained more experience in using the computer and the Internet as time progressed, which was expected. However, the percentage of students who reported having difficulty in using the computer increased from week 6 (9.7 percent) to week 10 (16.7 percent). Some possible reasons may have come from the fact that these students might rarely use a computer or that they might not have much time to use a computer because they had to deal with other courses at the same time. Thus, the results seemed to show that the computer experience might have come to be a factor influencing students' achievement. To explore this matter, the average of responses obtained by adding all numbers of responses was used to find out if there was any correlation.

Table 7.24: Correlation between computer usage experience from week 6 and the score of midterm examination

		Experience From Week 6	Midterm Score
Experience From Week 6	Pearson Correlation	1	.130
	Sig. (1-tailed)	.	.243
	N	31	31
Midterm Score	Pearson Correlation	.130	1
	Sig. (1-tailed)	.243	.
	N	31	39

Table 7.24 shows that there was no significant correlation coefficient between experience using a computer in week 6 and the midterm score ($r = 0.130$, $n = 31$,

$p > 0.05$, one-tailed). However, Table 7.25 shows that the correlation coefficient was significant between computer usage experience in week 10 and the final score ($r = 0.310$, $n = 32$, $p = < 0.05$, one-tailed). The results appear to show that when students had used the computer for 10 weeks, they were likely to have gained more experience and these results could be used to verify that computer usage had potential to be a factor influencing students' achievement as indicated by the correlation between the computer usage experience and the final score.

Table 7.25: Correlation between computer usage experience from week 10 and the score of final examination

		Experience Form Week 10	Final Score
Experience From Week 10	Pearson Correlation	1	.310*
	Sig. (1-tailed)	.	.042
	N	32	32
Final Score	Pearson Correlation	.310*	1
	Sig. (1-tailed)	.042	.
	N	32	39

* .Correlation is significant at the 0.05 level (1-tailed).

7.4.3 Conclusion

The results from this part of the study showed that students used DIVTIC much less in the second half of the course. The findings revealed that there was a correlation between the total of log-in times of students with a low GPA and the final examination score. This could be because students with a low GPA used DIVTIC significantly more than other students. This finding seems to suggest that time spent using DIVTIC was a factor influencing students' achievement.

As time progressed, students appeared to become experienced using DIVTIC and developed the ability to go straight to a particular point of DIVTIC to suit their needs. Thus, level of computer experience appeared to have the potential to be a factor influencing students' achievement. When students had used the computer and DIVTIC for extended periods of time, they gained more experience. This experience appeared to help them to navigate and use DIVTIC in a more effective way requiring less time to achieve.

CHAPTER 8: SUMMARY AND CONCLUSIONS

The thesis sought to explore the learning opportunities and advantages that could be gained through the use of contemporary multimedia technologies as instructional supports in introductory programming classes. The thesis involved the design, development and implementation of an interactive multimedia model called Dynamic Interactive Visualisation Tool in Teaching C (DIVTIC).

DIVTIC was designed by using multimedia and visual imagery to provide learners with a step-by-step representation of program executions in C language as a means to enhance their understanding. DIVTIC was designed to support knowledge construction and combined collaborative and visualisation learning strategies with use of the Internet and the World Wide Web to support the learning of introductory programming. The development of DIVTIC was based on contemporary learning theories applied using available communication technologies and the Internet as a delivery medium.

Designed to support learning introductory C programming, DIVTIC was comprised of eight components including:

1. Information of computer structure: A set of animations which was designed to explain each part of the computer to give students an overview of the basic structure of a computer and to provide the opportunity for students to become familiar with the overall functioning of a computer.
2. Syllabus/Lecture notes: A set of course materials and relevant information for students to download ahead of time during the trimester for an introductory C programming class.
3. Animated examples: A set of animation examples which students could interact with by clicking on the control buttons at anytime. The animations showed students each step of program execution of algorithm in C from the introduction to computer programming course. There were 46 animations in total. A marker was used to animate all the segments of each line of the program. Animation examples were divided into three different levels of

difficulty: (a) easy and short animations, (b) average difficulty animations, and (c) long and complex animations.

4. Notes about C compiler: This component contained a step-by-step animation that demonstrated how to use a C compiler. It was intended to help students become familiar with the C compiler and also to encourage them to write a simple program.
5. C references & links: This component contained information which assisted students in constructing their own knowledge by searching for relevant references on the server and the World Wide Web.
6. C WebBoard: This component allowed students to communicate with their peers. Students could post their questions and receive answers via the use of this feature.
7. Self-Evaluation exercises: This component was designed to provide an opportunity for students to test their own understanding or performance. It contained a set of multiple-choice questions which covered all topics in the computer programming course. It was designed to include a dynamic feedback window for students when they clicked on an answer.
8. FAQ pool: This component contained frequently asked questions (FAQs) by students in previous computer programming classes. This feature was intended to provide students with easy access to some common questions which peers had asked, together with teacher responses.

To explore how DIVTIC could influence learning, a study was designed and implemented with 100 undergraduate engineering students enrolled in Computer Programming 408101 at SUT in Thailand. The study used to explore the following research questions:

1. How do students use the DIVTIC?
   (1a) Which components of DIVTIC do students use and for how long?
   (1b) What strategies do students use with DIVTIC?
   (1c) What factors influence students' use of DIVTIC?
   (1d) What attitudes do students generate towards DIVTIC?

These questions sought to investigate how students in the study responded to this innovative tool in order that learning outcomes could be better understood. At the same time the study sought to examine how DIVTIC was used so that any implications for

broader use could be identified. A second set of research questions was investigated to explore the impact of DIVTIC on students' actual learning achievement.

2. To what extent does the dynamic interactive visualisation process implemented in DIVTIC influence learning outcomes?
   - (2a) How does the dynamic interactive visualisation process implemented in DIVTIC influence students' performance in programming?
   - (2b) How does use of the dynamic interactive visualisation process implemented in DIVTIC vary among students?
   - (2c) What levels and forms of cognitive engagement are evident among DIVTIC users?
   - (2d) What factors influence students' achievement with DIVTIC?

This chapter is designed to conclude the findings from both research questions and provide recommendations for future research.

8.1 Findings

The study used one group of 50 students as a control group (Group C) and another of 50 students as an experimental group (Group E) with the same teachers and tutors for both groups. Convenience sampling was used to select students by matching the GPA manually in choosing sample groups. Therefore, both groups were comprised of 50 students ranging from low to high GPA. Each group was divided into three different levels according to GPA: low—less than 1.78; average—1.78 to 2.22; and high—above 2.22. Group C, for example, included C1, C2, and C3 which referred to low, average, and high GPA respectively, and Group E was comprised of E1, E2, and E3 which also referred to low, average, and high GPA respectively. However, there were 17 students in Group C and 11 students in Group E who withdrew from the course. Thus, there were only 72 students (Group C = 33; Group E = 39) who participated in the entire study. The majority of students who withdrew from the course were those with a low GPA in both groups.

Both the experimental and control groups were treated in the same manner except for use of the DIVTIC system. The DIVTIC system was provided to be used only by students in the experimental group for about 30 to 45 minutes at the beginning of each laboratory session along with a weekly task for a trimester. The study was conducted in

normal classes with the researcher providing guidance to the participating teachers and acting as an observer in the classes when the experiment was conducted. The students were informed by their teachers of the study and the role of the observer.

In order to explore how students used DIVTIC, the following data was collected during the study:

- Strategies students used when interacting with DIVTIC;
- Length of time spent by students using DIVTIC,
- Place and time when students used DIVTIC;
- Levels of students' interaction with DIVTIC;
- Students' feedback on the design of DIVTIC;
- Students' impressions of DIVTIC as a learning tool;
- Students' attitudes from using DIVTIC;
- Students' motivation in using DIVTIC;
- Students' satisfaction with DIVTIC
- Tutors' impression of DIVTIC as a learning tool;
- Factors influencing students' use of DIVTIC (time, content, and networking);
- Components of DIVTIC used by students and for how long; and
- Log files describing Web pages accessed by students.

This data was collected from such sources as screen recordings, log files, tutor interviews, student interviews, student questionnaires, and researcher observation.

To explore the impact of DIVTIC on students' programming performance, the following data was collected:

- Students' performances from the laboratory tests among the control and experimental group;
- Student's performances from the formal midterm and final examinations among the control and experimental groups;
- Performances of students with different learning abilities (low, average, and high GPA)

- Students' attempts in completing the weekly tasks;
- Forms and levels of interaction of students with different learning abilities (low, average, and high GPA); and
- Length of time spent by students with different learning abilities (low, average, and high GPA).

8.1.1 Research Question 1: How do students use DIVTIC?

The outcomes from analysis of the empirical data provided some very interesting and, in some cases, unexpected findings. The following section provides a summary and discussion of the findings which can be generated from this study.

- **Students prefer the animated examples component over all others in DIVTIC**

DIVTIC was comprised of eight components: Syllabus/Lecture Notes, Computer Structure, Animated Examples, C Compiler, C WebBoard, Self-assessment, FAQ Pool, and C References and Links. The animated examples component was designed to be a major component that provided animations showing students each step of program execution. At the beginning of each laboratory session, students were given a weekly task sheet requiring them to predict outcomes and to use DIVTIC to check their answers by viewing an associated animation. This process was designed to provide structure to the use of DIVTIC and to encourage students' use in a meaningful and deliberate fashion.

The empirical data from the log-in records revealed that the animated examples component was the most often-used component and followed by the Syllabus and lecture notes, the Computer structure, the C compiler, the Self-evaluation, the C references and links, the FAQ pool, and the C WebBoard.

The animated examples component was designed to help students to visualise what was happening when a program was executed step-by-step. Although, students were required to use the animated examples component along with the weekly task, they had opportunities to use other components to explore what they needed at any time. Students seemed to prefer the animated examples component over all others in DIVTIC. This may be because they were given the weekly task sheet to complete and asked to view an associated animation in the animated examples component to check their answer. These

findings suggest that the weekly task could be a major instrument that could be used to motivate students to use all parts of DIVTIC. A weekly task with more broadly based activities could possibly create the opportunities for students to use all of the DIVTIC components in meaningful and effective ways.

- **Without teacher direction, the C WebBoard component will not likely be used by students using DIVTIC**

The C WebBoard component was designed as part of learning environment. This component was intended to provide opportunities for students to communicate with their peers. This feature would encourage individuals to share and change their ideas leading to an opportunity to discover, analyse, synthesise, and evaluate each others' thoughts. It was intended that students could post their questions and receive answers via the use of this feature. However, the findings revealed that students in this study did not use this component at all. Some possible reasons may have come from the fact that these students used to a traditional learning style when information is transmitted, and not familiar with the learning strategy involving knowledge construction. It may have been that some were too shy to ask questions of their peers via the use of the WebBoard since they already had a chance to discuss issues and concerns in the laboratory sessions.

The findings suggest that teachers using DIVTIC need to discover ways to engage students to use the WebBoard as a channel to communicate, share ideas, or get feedback from peers or the teachers themselves. For example, teachers could provide a weekly topic posted on the WebBoard and ask students to respond to that topic by rewarding them with a minimal mark towards their total score, as an encouragement. This could motivate students to participate with the WebBoard in order to expand their knowledge and also have a chance to gain some marks.

- **Students with an average or high GPA are more likely to test their understanding than low GPA students**

DIVTIC also provided an opportunity for students to test their understanding via the use of the self-assessment component which was comprised of a set of multiple-choice questions that covered all topics. This component provided a dynamic feedback window for students when they clicked on an answer. This was intended to challenge students to participate and improve their learning outcomes.

The findings revealed that students with an average or high GPA seemed more likely to test their understanding via the use of the self-assessment component than those with a low GPA. This was an expected result. Students with an average or high GPA made faster progress in learning than those with a low GPA so that they had more time during the laboratory session to test their understanding via the use of the self-assessment component.

The findings suggest that teachers using DIVTIC may motivate students with a low GPA to test their own understanding through use of the self-assessment component by, for example, keeping records of students who used this component and rewarding them in some way. Generally, one would imagine that when students test their understanding with strategies that provide immediate feedback over a period of time, they would eventually gain some knowledge and learning advancement. Such a method could be used to encourage students with a low GPA to use the self-assessment component more since they could get some academic credit from using this component and would also gain some knowledge at the same time.

- **Students tend to see little value in the use of the FAQ pool and the C references and links components**

DIVTIC included a FAQ pool and C references and links components as part of the learning environment. The FAQ pool component was intended to be a knowledge-based pool that contained frequently asked questions (FAQs). This feature provided students with easy access to some common questions that peers have asked, together with answers. The C references and links component was a kind of information pool, which was intended to assist students in constructing their own knowledge by providing them with access to relevant information on the server and the World Wide Web.

The findings revealed that these two components were perceived to be of minimal value among students and they did not frequently use these components. Some students did not even bother to use these either. These two components were included to enable students to explore their knowledge and available related information. Students are likely to use these components whenever they need to seek further information.

The findings suggest that teachers using DIVTIC may include information that students would want to access, such as frequently asked questions relating to topics relevant to

either the midterm or final examinations. Teachers may announce this feature to students in the initial weeks to ensure that students know the benefit from using these two components. This may be a strategy which encourages students to make more use of these two components.

- **Students appear to have more interactions toward the end of the course**

DIVTIC was designed to provide capability for students to control the animation process by including a set of control buttons that student could interact with. By interacting with DIVTIC, students can access information in a non-linear fashion that can come from any exploration paths. This capability was designed to enable students to monitor and manage their own learning and to construct better understanding in programming concepts and as a result to gain more motivation for learning.

The data from the questionnaires revealed that students in this study seemed likely to have more interactions with DIVTIC as the course progressed. They seemed more likely to use the control buttons to interact with DIVTIC at the end of the course than at the beginning. A possible reason may have been that students may have found this way to interact with DIVTIC only after an extended period of time.

Students could use the control buttons to stop and pause which would enable them to pause and think, compare, or reflect on their thoughts. This feature was expected to be used consistently throughout the study. A possible explanation for low use early in the study could have been that students gained more experience using DIVTIC as the course progressed. The findings suggest that teachers using DIVTIC should demonstrate how to interact with DIVTIC fully in the very first weeks in order to allow students to get used to the controls and to know exactly what they could do with the control buttons.

- **The language used in DIVTIC appears to be a factor that impede students' use of DIVTIC**

The study was undertaken at Suranaree University of Technology (SUT). Students in this study typically had lower abilities in learning than students in other government universities in Thailand. This is because SUT is a new university established in 1990 and the first cohort of students was admitted in 1993 academic year. Most of the

students, who took a standard entrance examination and admitted into SUT, had selected SUT as their last choice.

DIVTIC was designed to be a supplementary tool to help students learn programming. It provided a Message Board panel that displayed related information as the animation progressed. The information was written in English. In this study, the difficulty in understanding the English explanations in the Message Board panel for each animation seemed to be a significant factor in obstructing students' use of DIVTIC, since their native language was Thai. The study confirmed that when students had to learn both programming concepts and English at the same time, they understandably were not to be able to handle both and some words seemed likely to mislead their understanding. This was an unexpected result because most of the English explanations used in DIVTIC were technical terms. However, students in this study had low English skills and needed more time to acquire the information they needed. If DIVTIC had been used with students for whom English was their native language, the outcomes may have been different.

The findings suggest that DIVTIC may need to be modified from English explanations to Thai, with the inclusion of audio explanations instead of using static text. This may help students save time through listening rather than reading all the explanations. If English explanations were used, teachers using DIVTIC may have needed to translate the explanations into their students' native language to prevent any misunderstandings or misconceptions.

- **Students tend to respond well to the use of DIVTIC as a complement to programming learning**

Students in this study were given questionnaires twice during the study. They were the same questionnaire but given at different times. One was given in week 6 and another one was given in week 10. The questionnaire sought their feedback on issues associated with use of DIVTIC as a complement to learning.

The study revealed that students responded very positively to the use of DIVTIC as a complement to programming learning and the fact that they were able to freely navigate through the resource they needed at their own pace and in their own time. They claimed to have gained confidence from using DIVTIC to solve basic programming problems.

Most of the students seemed to be satisfied and interested in using the animated examples component. The interface of the animated examples was found to be comfortable and user-friendly tool, easy to use and navigate. The animated examples component was felt to have covered all aspects appropriately and provided concepts that were well addressed and well explained. Most of the students seemed more likely to use this component as a group but they may have been too shy to communicate with their peers.

The findings suggest that the DIVTIC system could be used by teachers and students in a similar setting to:

- encourage and support the majority of students to think along with the animation process so that they could reflect on what they were trying to get out of it;
- help students to solve more complicated tasks or enable them to help their peers as the course progressed;
- encourage students to pay more attention in class;
- help students create learning outcomes more efficiently;
- help students gain more enjoyment and feel more comfortable;
- entertain the students in their learning process;
- assist in learning to program;
- help students apply concepts to solve given problems and also enhance their understanding; and
- help students to learn the programming environment and program execution.

8.1.2 Research Question 2: To what extent does the dynamic interactive visualisation process implemented in DIVTIC influence learning outcomes?

The outcomes from analysis of the empirical data provided some very interesting and, in some cases, unexpected findings. The following section provides a summary and discussion of the findings which can be generated from this study.

- **The use of DIVTIC can significantly increase low GPA students' programming achievement**

The findings from this study clearly demonstrated that when low GPA students used DIVTIC their programming achievement was significantly enhanced over similar students whose learning did not involve this tool. In comparisons between the achievement of the students in the control and experimental groups, the achievements of students with a low GPA were compared.

The study revealed that students with a low GPA in the experimental group had a higher mean in each test than those in the control group, especially in the midterm and final examinations which were obviously relatively large differences. The findings suggest that students may not have been using DIVTIC long enough by laboratory test 1 for the tool to make a difference. However, continued use throughout the trimester was found to contribute significantly to students' programming performance.

DIVTIC was designed to help students to visualise and conceptualise programming constructs through their interaction with a tool that helped them to focus and engage with important steps and processes in the solution of programming algorithms. The study confirmed that a visualisation process could be of considerable assistance to a particular group of students, those with a low GPA, in developing their understanding of difficult programming concepts. The lower GPA students could view the process in a visual fashion helping their understanding. The findings suggest that DIVTIC would be a useful tool for teachers looking to promote understanding among their learners with a low GPA.

- **The use of DIVTIC can impede the programming achievement of average or high GPA students**

The outcomes from this study revealed an unexpected finding in terms of the contribution made by DIVTIC to students with an average or high GPA. The study found that among those students were made to use DIVTIC, programming achievement as demonstrated by results in the tests, actually diminished compared to their counterparts in the control group who did not use DIVTIC.

The scores of each test of students in both experimental and control were used to explore if there were any significant differences between each level of students'

learning abilities, low, average, or high GPA. The findings revealed that students with an average or high GPA did not improve their achievement through the use of DIVTIC. Although, DIVTIC was designed to help all students, use of DIVTIC was found to impede the programming achievement of those with higher learning abilities.

This was an unexpected finding. One would imagine that if such an interaction did not help their learning, it certainly should not impede their programming achievement. The cause of the impediment appeared to come from how the students were required to use DIVTIC rather than from use of the program itself. Some possible explanations may have come from the fact that the visualisation process in DIVTIC may not have been challenging or been appropriate to students with an average or high GPA. Another possibility is that the problems may have needed to demonstrate more complex concepts. The most likely reason for the impeded learning appears to come from the time students spent using DIVTIC. The use of DIVTIC along with the weekly tasks was very time consuming and may have taken away from students' time spent solving weekly problems given by the teacher. Perhaps, students with an average or high GPA may have already understood the basic concepts, and did not need to invest their time using DIVTIC to discover what they already knew.

The findings suggest that the mandatory use of DIVTIC for such periods of time without any exception may not benefit all students especially those with an average or high GPA. This seems to suggest that the use of DIVTIC by teachers should be flexible and made when students are found needing in their understanding.

- **Students with a low GPA tend to use DIVTIC more than others**

DIVTIC was planned to be used, most of the time, by students who may have had difficulty in understanding abstract programming concepts. All students in the experimental group were given the same opportunity and the same tasks to complete with DIVTIC. The results clearly showed that those students with a low GPA took more time and completed more exercises than others. Some possible reasons could be that those students with an average or high GPA might have already understood the basic programming concepts so that they used DIVTIC less. There was considerable repetition in the animations. Students with an average or high GPA may have gained the concepts before those with the low GPA. Students with higher GPA seemed likely to

make faster progress and to require less time to complete the weekly tasks with DIVTIC.

This finding suggests that the low GPA students saw greater benefit in the use of DIVTIC than others and were encouraged by their successes. On the other hand, those students with an average or high GPA must not have seen the benefits to be derived from the use of DIVTIC and were less inclined to continue using it.

- **Among students with a low GPA, high levels of use of DIVTIC correlates with higher programming achievement**

The results from this study revealed that there were differences in the time spent using DIVTIC among students with a low GPA. The results showed that students who used it more were more likely to have higher programming achievement than other students who used it less. This was a promising outcome. Generally, one would imagine that students with a low GPA would be expected to use the tool to help their understanding in programming concepts and the more time they spent using DIVTIC, the higher programming achievement they would achieve. The outcomes confirm this expectation.

This finding suggests that teachers using DIVTIC should encourage and motivate students with a low GPA to use DIVTIC as much as possible. Higher levels of use can contribute to the development of students' conceptual understanding of programming language.

- **The form of interaction with DIVTIC does not necessarily influence programming achievement**

DIVTIC was designed to be an interactive tool which provided students with the capacity in control the animation. This feature was intended to enable students to conceptualise and reflect throughout the animation process by pressing the control buttons including *Stop*, *Play*, *Step Forward*, *Step Backward*, *Go to the Beginning*, and *Go to the End* buttons. The findings revealed that most of the students who were highly interactive with the use of DIVTIC were those with an average or high GPA. However, their interaction did not appear to correlate with increases in their programming achievement.

This was an unexpected result because this form of interaction was intended to help and engage learners in a process involving higher order thinking that should have enabled

them to gain understanding. Some possible explanations may have come from the fact that these students made faster learning progress and had more time to interact with the functionality of DIVTIC than those with a low GPA. Without watching the entire animation process from the starting point to the end, students possibly might not develop the intended understandings of the programming process.

The findings suggest that teachers using DIVTIC should guide students at the initial stage to watch the entire animation process before using the control buttons to interact with DIVTIC. Once, students have seen through the animation process, they could then be free to play around with the control buttons. This seems likely to be able to prevent any misunderstanding in any parts of the source code. Another possibility to encourage students to view the entire program, would be to have DIVTIC provide only the *Play* and *Stop* buttons at the initial stage. The other buttons, *Step Forward*, *Step Backward*, *Go to the Beginning*, and *Go to the End* buttons, may be included when the animation process had run completely once.

- **Lower GPA students appear to learn from non-interactive use of DIVTIC**

DIVTIC was designed to provide students with the capacity in control the animation process. One would imagine that the lower GPA students, who seemed likely to have more difficulties in understanding programming concepts, might have higher interactions with the tool than other students. This interaction could help them to focus and engage with important steps and processes in the solution of programming algorithms. However, lower GPA students in this study appeared to learn from simply viewing the animations rather than being highly interactive and stopping and starting them consistently.

The successful use of DIVTIC seemed not always to require students to be highly interactive. Students with a low GPA watched the animation process all the way through with less interaction. This strategic use of DIVTIC took more time. However, these students seemed likely to gain better understanding by spending more time in just viewing the animation process itself.

The findings suggest that teachers using DIVTIC should require all students to watch the animation process all the way through at least once before reviewing or interacting. Teachers may also demonstrate the entire animation process in the class once before

going to the laboratory. This seems likely to help students to gain some ideas by observing what is going on when the program executes at any particular step before they actually try it themselves.

- **Over extended periods of time, students are able to gain learning advancement from reduced use of DIVTIC**

All students in the experimental group were required to use DIVTIC along with the weekly task at the beginning of each laboratory session for 30 to 45 minutes. They also were encouraged to use DIVTIC at their own pace and in their own time. The data revealed that even though students tended to use DIVTIC less as the course progressed, they achieved the same level of problem success.

When students had used DIVTIC over extended periods of time, they gained experience on how to use the feature and functionality provided in such a tool. Thus, they could just go straight to a particular point of the tool to suit their needs without wasting their time on unnecessary features. This experience appeared to help them to navigate and use DIVTIC in a more effective way with less time to achieve success. Interestingly, the lower GPA students also made less use of DIVTIC as time progressed, but they still got as much done.

The findings suggest that teachers using DIVTIC may provide extra tasks relating to some examples in DIVTIC for student to practise as the course progressed. This can encourage students to be active learners by giving them opportunities to solve such tasks, and this can also motivate students to use DIVTIC more to compare their results or when they face difficult steps in solving programming tasks.

- **Time spent using DIVTIC is a factor that appears to influence students' programming achievement**

Students in the experiment group were required to use DIVTIC at least once a week during the laboratory session. There were differences between time spent using DIVTIC among students which appeared to influence students' programming achievement. The results from this study revealed that time spent using DIVTIC among all students in the first stage (weeks 2 to 6) correlated with the score of the laboratory test 1 undertaken in week 7. However, there was no correlation between the total of log-in times and the score of the final examination.

A further exploration into each group of students' log-in times was carried out to see if there were any correlations between time spent using DIVTIC of each group and the scores of each test. The findings revealed that there was a correlation between the total log-in times of students with a low GPA and the final examination score. This could be because students with a low GPA used DIVTIC significantly more than other students.

The findings seem to suggest that time spent using DIVTIC is a factor influencing students' programming achievement. This was a promising result because it showed the more time students spent using DIVTIC, the higher outcomes they could achieve. Teachers using DIVTIC may encourage students with a low GPA to use DIVTIC as much as possible. On the other hand, students with an average or high GPA might best be provided with more challenging tasks relating to examples in DIVTIC as their course progresses. This could motivate them to use DIVTIC consistently and may help them to achieve higher programming achievement. The findings suggest the need for teachers to adopt DIVTIC in flexible ways to cater for the varying needs of their programming students.

8.2 Suggested Improvements for DIVTIC

The overall findings from the study have suggested numbers of possible ways to improve DIVTIC and the way it is used with students, in order to improve its capacity to influence learning. From a technical perspective, the detailed study of DIVTIC has suggested that for the target audience, DIVTIC would be enhanced by:

- **Instructing and description in Thai:** The explanation in the Message Board panel of each animation could be translated into Thai. This would help Thai students who were weak in English to understand better.
- **Viewing through the entire animation process:** Students should view the animation process all the way through at least once before reviewing or interacting with DIVTIC. This could ensure that students would not miss any important steps and processes in the solution of programming algorithms.
- **Opportunity for students to test their own input:** DIVTIC may be designed to provide an opportunity for students to be able to test their own input beside the given input from DIVTIC so that it makes learning meaningful and students may have more interest. One suggestion from a user suggested that it

would be more appropriate and interesting if they could test their own input via the use of DIVTIC.

From an implementation perspective, use of DIVTIC might best be achieved through learning setting which exhibits the following characteristics:

- **More time for students:** The two-hour laboratory session time in this study appeared to be inadequate. Students clearly need more time to achieve abstract programming concepts. Incorporating DIVTIC into a schedule requires students to spend more time and this would need to be built into the setting.
- **More flexible approaches to its use:** Students should be required to use DIVTIC for differing amounts of time depending on how much each student needs. Low GPA students could be encouraged to use DIVTIC as much as they can, while students with an average or high GPA might need less time to achieve the same success.
- **DIVTIC on CDs:** The DIVTIC system may be produced on a CD and distributed to all students to install the software onto their own computers. This could reduce network problems often evident in Thailand.
- **More advanced programming concepts:** DIVTIC was used as a supplementary tool to help students in learning how to program at SUT. It covered only basic C syntax since the 408101 course, *Computer Programming in C*, was planned and prepared to teach basic introduction to computer programming to engineering students. It may be of more benefit to include more advanced C syntax so that DIVTIC could be used to support all levels of student abilities in learning.

8.3 Potential Limitations of the Study

As Patton (1990) notes "There are no perfect research designs. There are always trade-offs" (p. 162). The study sought to explore how the students in an introductory programming course were aided by the use of interactive visual and Web-based instructional materials. There were a number of factors likely to limit the generalisability such as:

- **Limited learning setting:** This thesis reports on a single university studying at Suranaree University of Technology (SUT). Students at SUT are more likely to live in a rural area and have a low level of intake. These students have low English skills comparing to other government universities in Thailand, as they use Thai language most of the time in their everyday lives. There may have been cultural and social issues associated with the Thai setting that have influenced findings that may not be evident in other settings.
- **Language of implementation:** Students at SUT speak and write in Thai, their native language. They study using Thai textbook in all their courses. Thus, having a text written in English as in this study may have caused some difficulties in their learning process. DIVTIC was designed and developed using English. Only the problems were written in Thai. However, the meaning and description of each keyword and the explanation of each animation process were written in English. Difficulties with English may have caused difficulties with the use of DIVTIC which may have limited the findings.
- **Nonrandom sampling:** By using convenience sampling to select students for the control and experimental groups, the students in both groups were not necessarily representative of the population. Sproull (1988) argues that results from nonrandom sampling cannot be generalised because they have usually not been defined. The findings would have been more generalisable if the student sampling had come from a randomised process.
- **Variation in learner:** There were many variations in learners. A more homogeneous group of students would have provided more opportunity to explore how best to use DIVTIC. If the students in this study had the same levels of abilities in learning, the findings may have been generated in a different way.
- **Sample size:** The students in this study were able to withdraw from the course anytime before the withdrawal penalty period. It was not guaranteed that all students in both groups would remain throughout the study period. There were 33 out of 50 students who participated in the control group, while 39 out of 50 students participated in the experimental group. The results from this small number of students may be difficult to generalise.

- **Duration of intervention:** The time that students used DIVTIC was limited for 30 to 45 minutes depending on how easy or difficult the weekly task was in each laboratory session. The tutors sometimes spent an extra 30 minutes for teaching how to solve the instructor's weekly problem, and left the remaining time for the students to write the C source code to solve that problem and test it via the C compiler. The two-hour laboratory seemed to be insufficient for students to complete all the tasks, weekly task and instructor's weekly problem. A four-hour laboratory would be more appropriate for students learning to programming through the use of DIVTIC. The lack of time with DIVTIC, caused by environmental factors outside the control of the study, may have limited the findings.
- **Lack of pretest.** This study did not provide a pretest in both the control and experimental groups. The students in both groups may not have been matched perfectly in term of their learning abilities and achievement. Although, the students' GPA in both groups were matched perfectly by using convenience sampling, those GPA were acquired from the first trimester which students enrolled only in basic fundamental courses. The GPA from those courses may not have been able to match the ability of each subject perfectly for the purpose of this study, a computer programming course, which was mainly involved with mathematical and scientific matters.
- **Short time span:** This study was conducted over one trimester that was comprised of 12 weeks for lecture plus another week, the 13th week, for the final examination period. The time that students used DIVTIC was limited to 30 to 45 minutes depending on how easy or difficult the weekly task was in each laboratory session. A longer period of time in using DIVTIC might have led to greater changes in students' achievement than appeared in this study.
- **Novelty of learning design:** This study was undertaken with students who were accustomed to learning in a traditional style, including direct teaching and knowledge transmission. In such settings, teachers provide students with knowledge to be memorised and repeated usually without providing opportunities for them to make sense of the information they have been given. DIVTIC was designed to support student-centred learning which engages students as active learners by thinking along with and interacting with the

animation process. This would have been an unusual learning style for the students, which they might have needed time to get used to.

The study may have been able to show more significant results if students had been familiar with student-centred learning and better able to choose how and when to use a tool such as DIVTIC.

8.4 Suggestions for Future Research

Further study is needed to establish precisely the design of elements of interactive multimedia, which encourage and enable students with an average or high GPA to achieve their ultimate outcomes. Furthermore, the two-hour laboratory session did not seem to be adequate. Novice students may need more time to finish both DIVTIC's weekly task and the instructor's weekly problem with tutor's assistance. In some cases, if the Internet connection possibly is a problem suggesting that the DIVTIC system could be produced on a CD and given to all students.

The findings from the study provide strong support for the concept of a dynamic and visual programming aid such as DIVTIC as a support for introductory programmers. A number of areas emerged from the study as potentially important areas of inquiry to further explore ways to maximise learning from such tools. Possible areas of further inquiry include:

- **Using DIVTIC for more complex problems**

Students with an average or high GPA may have already understood some basic programming concepts. Teachers using DIVTIC may provide more complex problems to ensure that students with an average or high GPA are satisfied in using DIVTIC and see some benefits. In this study, DIVTIC was created to support the learning of introductory programming concepts. It could also be used to support the learning of more advanced programming concepts. Further research into the efficacy of such a tool supporting the learning of high achieving students would be useful to explore the full extent of the possibilities of this learning support.

- **Encouraging particular patterns of use:**

This study found that students used DIVTIC in a variety of forms and for varying times. Students often failed to view the animation in their entirety. It would be useful in further research to explore different usage patterns to determine optimal forms for teachers to encourage. In such research, teachers using DIVTIC could encourage students to view the entire animation process at least once before interacting with DIVTIC. This could ensure that students would, at least, see each entire animation once. The findings suggest this could enhance learning but further research is needed to explore and confirm this possibility.

- **Using DIVTIC in other programming languages**

Although DIVTIC was designed to help students in learning how to program in C, students learning other programming language could also use DIVTIC to help them understand basic programming concepts. Most programming languages have the same concepts and algorithms. Further research could explore the reusability of DIVTIC in other programming domains and could establish strategies for facilitating the design of subsequent systems in ways that support this form of application and reuse.

- **Exploring more flexible ways to use DIVTIC in classrooms**

Learning theory suggests that students should have the opportunity to use tools such as DIVTIC on their own terms. They should have flexibility in their use of DIVTIC. Students with a low GPA may need more time than those with an average or high GPA. Students with a low GPA could be encouraged to use DIVTIC as much as they can, while other students could be free to use DIVTIC as they choose. It would be helpful in determining how best to implement a tool like DIVTIC to explore learning outcomes from flexible approaches. Further research that could assist in this process would involve applications of DIVTIC with varying forms of teacher support among diverse students groups.

- **Alternative delivery strategies**

In the study, DIVTIC was provided to students through a Web-based delivery. This was limiting in a number of respects. It limited the levels of student access and limited places from which access could be gained. Given the nature of the animation section of the tool, there are other more flexible ways that could be used to provide students access to this tool. For example, DIVTIC could be uploaded onto a server and also produced

on a CD to distribute to students. This could ensure that if there was an Internet connection problem, students would still be able to use DIVTIC on their own machine. Also production of such learning resources on CD can provide cost savings. Useful further research could explore possible forms of CD delivery for various components of DIVTIC to explore how this might improve access and lead to higher levels of incidental access. Discovering the optimal delivery form for the components of DIVTIC would help to improve learning outcomes from its implementation.

8.5 Conclusion

The results from this quantitative study have provided strong support for the notion that use of DIVTIC can assist novices in learning introductory computer programming. The results are interesting that they clearly demonstrate the advantage of DIVTIC with students with a low GPA. What are the learner characteristics inherent in this group? Is it low formal reasoning or poor self-regulation? If we could identify the learner characteristic, we could use DIVTIC more widely. However, the study has suggested that DIVTIC is a valuable tool for novice programmers and encourages further exploration and inquiry. This model seems to have potential over traditional face-to-face teaching and it is a strong complement for teaching and learning innovations and initiatives in introductory computer programming courses.

CHAPTER 9: REFERENCES

Aiken, L. R. (1997). *Questionnaires and inventories: Surveying opinions and assessing personality.* New York: John Wiley & Sons, Inc.

Alam, J., & Renci, J. J. (1998). Use of Internet in information content creation and delivery for promoting active cooperative learning. *Computers in Education Journal, 8*(1), 23-26.

Albion, P. (2000, July). *Developing interactive multimedia using a problem-based learning framework.* Paper presented at the ASET-HERDSA Conference 2000, University of Southern Queensland, Toowoomba, Australia.

Allison, B., O'Sullivan, T., Owen, A., Rice, J., Rothwell, A., & Saunders, C. (1996). *Research skills for students.* London: Kogan Page Limited.

Astrachan, O., & Rodger, S. H. (1998, February). *Animation, visualization, and interaction in CS1 assignment.* Paper presented at the Proceeding of the Twenty-ninth SIGSCE 98 Technical Symposium on Computer Science Education, Atlanta, Georgia, USA.

Azemi, A. (1995, September 29). *Teaching computer programming courses in a computer laboratory environment.* Retrieved June 14, 2000, from http://fre.www.ecn.purdue.edu/FrE/asee/fie95/2a5/2a55/2a55.htm

Bagui, S. (1998). Reasons for increased learning using multimedia. *Journal of Educational Multimedia and Hypermedia, 7*(1), 3-18.

Baldwin, L. P., & Macredie, R. D. (1999). Beginner and programming: Insights from second language learning and teaching. *Educational and Information Technologies, 4*(2), 167-179.

Bayman, P., & Mayer, R. E. (1988). Using conceptual models to teach BASIC computer programming. *Journal of Educational Psychology, 80*(3), 291-298.

Bazik, J., Tamassia, R., Reiss, S. P., & Van Dam, A. (1998). Software visualization in teaching at Brown University. In J. Stasko, J. Domingue, M. H. Brown & B. A. Price (Eds.), *Software Visualization: Programming as a multimedia experience* (pp. 383-398). Cambridge, Massachusetts: MIT Press.

Bell, J. (1993). *Doing your research project: A guide for first-time researchers in education and social science* (2nd ed.). Buckingham, Philadelphia: Open University Press.

Bergin, J., Brodlie, K., Goldweber, M., Jimenez-Peris, R., Khuri, S., Patino-Martinez, M., et al. (1996, June). *An overview of visualization: Its use and design.* Paper presented at the Proceedings of the 1st Conference on Integrating Technology into Computer Science Education, Barcelona, Spain.

Bishop-Clark, C. (1995). Cognitive style and its effect on the stages of programming. *Journal of Research on Computing in Education, 27*(4), 373-386.

Boatwright, J. R., & Slate, J. R. (2000). Work ethic measurement of vocational students in Georgia. *Journal of Vocational Education Research, 25*(4), 113-141.

Brilliant, S. S., & Wiseman, T. R. (1996, February). *The first programming paradigm and language dilemma.* Paper presented at the SIGCSE'96, Philadelphia, USA.

Brock, P. A. (1994). *Educational technology in the classroom.* Englewood Cliffs, New Jersey: Educational Technology Publications, Inc.

Brooks, D. W. (1997). *Web-teaching: A guide to design interactive teaching for the World Wide Web.* New York: Plenum Press.

Brown, J., & Duguid, P. (1993). Stolen knowledge. *Educational Technology, 33*(3), 10-15.

Brown, T. (1997). *Multimedia in education: Multimedia.* Retrieved September 16, 2002, from http://scs.une.edu.au/Materials/573/573_11.html

Bruffee, A. K. (1995). Sharing our toys: Cooperative learning versus collaborative learning. *Change, 27*(1), 12-19.

Brummund, P. (1997). *Educational animations.* Retrieved September 23, 2000, from http://www.cs.hope.edu/~dershem/reu/papers97/brummund/pwb.html

Brusilovsky, P. (2001, October). *WebEx: Learning from examples in a programming course.* Paper presented at the WebNet' 2001: World Conference of the WWW and Internet, Orlando, USA.

Bryman, A. (1988). *Quantity and quality in social research.* London: Unwin Hyman.

Bryman, A. (1989). *Research methods and organization studies.* NSW, Australia: Allen & Unwin (Australia) Limited.

Burns, B. R. (1994). *Introduction to research methods* (2nd ed.). Melbourne, Australia: Longman Cheshire.

Butcher, D. (n.d.). *An overview of learning theory with an emphasis on problem based learning.* Retrieved August 29, 2000, from http://www.usd.edu/~knorum/learningpapers/pbl.html

Cardinale, L. A. (1991). Conceptual models and user interaction with computer. *Computers in Human Behavior, 7*(3), 163-169.

Carlos, F. M. (1998). *Collaborative learning and architecture education.* Retrieved September 14, 2000, from http://ucs.orst.edu/~healeyd/s60/carlos.html

Carter, J., & Jenkins, T. (1999, June). *Gender and programming: What's going on?* Paper presented at the 4th Annual SIGCSE/SIGCUE on Innovation and Technology in Computer Science Education, Krakow, Poland.

Cobb, T. (1999). Applying constructivism: A test for the learner-as-scientist. *Educational Technology Research and Development, 47*(3), 15-31.

Collins, A., & Brown, J. S. (1988). The computer as a tool for learning through reflection. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems* (pp. 1-18). New York: Springer-Verlag.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453-494). Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.

Computers-in-use in year 2000. (2000). Retrieved August 11, 2000, from http://www.c-i-a.com/19981103.htm

Corderoy, R. M., & Lefoe, G. (1998, October). *Designing and developing effective interactive Web environments for teaching and learning.* Paper presented at the Proceeding of ICCE'98: Global Education on the Net, Beijing, China.

Creswell, J. W. (1994). *Research design: Qualitative & quantitative approaches.* Thousand Oaks, California: Sage Publications, Inc.

Crynes, B. L., & Hawley, J. A. (1995). *Professor silicon and professor maestro-The perfect combination.* Retrieved September 16, 2002, from http://fie.engrng.pitt.edu/fie95/2a3/2a33/2a33.htm

Cunningham, D. J., Duffy, T. M., & Knuth., R. A. (1993). Textbook of the future. In C. McKnight, A. Dillon & J. Richardson (Eds.), *Hypertext: A psychological perspective* (pp. 19-49). London: Ellis Horwood Limited.

Dalton, D. W., Hannafin, M. J., & Hooper, S. (1989). Effects of individual and cooperative computer-assisted instruction on student performance and attitudes. *Educational Technology Research and Development, 37*(2), 15-24.

Daly, C. (1999, June). *RoboProf and an introductory computer programming course.* Paper presented at the 4th Annual SIGCSE/SIGCUE on Innovation and Technology in Computer Science Education, Krakow, Poland.

Deacon, H., Walton, M., & Wilson, F. (1997). *Developing multimedia for higher education: The experience of the Isiseko project.* Retrieved September 16, 2002, from http://www.meg.uct.ac.za/papers/fwpaper.htm

Denyer, G. (1997). *Application of explanation based learning to the teaching of biochemical calculations.* Retrieved August 31, 2000, from http://science.uniserve.edu.au/newsletter/vol8/denyer.html

Denzin, N. K., & Lincoln, Y. S. (1998). Introduction: Entering the field of qualitative research. In N. K. Denzin & Y. S. Lincoln (Eds.), *Strategies of qualitative inquiry* (pp. 1-34). Thousand Oaks, California: Sage Publications, Inc.

Dershem, H. L., & Brummund, P. (1998, March). *Tools for Web-based sorting animation.* Paper presented at the SIGSCE'98, Atlanta, USA.

Dershem, H. L., & Vanderhyde, J. (1998, February). *Java class visualization for teaching object-oriented concepts.* Paper presented at the Proceeding of the Twenty-ninth SIGSCE 98 Technical Symposium on Computer Science Education, Atlanta, USA.

Doolittle, P. E. (1998, April). *Integrating constructivism & cognitivism.* Paper presented at the Ninth International Conference on College Teaching and Learning, Jacksonville, Florida, USA.

du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research, 2(1)*, 57-73.

Dyck, J. L., & Mayer, R. E. (1989). Teaching for transfer of computer program comprehension skill. *Journal of Educational Psychology, 81*(1), 16-24.

Dyrli, O. E., & Kinnaman, D. E. (1995). Part I: Technology in education: Getting the upper hand. *Technology & Learning, 15*(4), 38-43.

Enerson, D. M., Johnson, R. N., Milner, S., & Plank, K. M. (1997). *Teaching with collaborative activities and small groups.* Retrieved September 13, 2000, from http://www.psu.edu/celt/PST/collab1.html

Ertl, B., & Kraan, A. G. (1997, June). *Internet-based learning environments from a constructivist point of view.* Paper presented at the RILW - Romanian Internet Learning Workshop, Ilieni, Romania.

Erzberger, C., & Prrein, G. (1997). Triangulation: Validity and empirically-based hypothesis construction. *Quality & Quantity, 31*, 141-154.

Essaka, M. F. (1998, September 4, 2000). *PowerPoint class.* Retrieved September 20, 2002, from http://www2.umist.ac.uk/isd/lwt/apt/ppclass/home.htm

Fardouly, N. (1998a, December 23). *Principles of instructional design and adult learning: Student-centred teaching strategies.* Retrieved August 29, 2000, from http://www.fbe.unsw.edu.au/learning/instructionaldesign/strategies.htm

Fardouly, N. (1998b, September 15, 1999). *Teaching with the Web: Instructivist versus constructivist learning.* Retrieved September 29, 2000, from http://www.fbe.unsw.edu.au/learning/teach-with-web/analysis/inst-const.htm

Fay, A. L., & Mayer, R. E. (1994). Benefits of teaching design skills before teaching Logo computer programming: Evidence for syntax-independent learning. *Journal of Educational Computing Research, 11*(3), 187-210.

Fetherston, T. (1998). A socio-cognitive framework for researching learning with IMM. *Australian Journal of Educational Technology, 14*(2), 98-106.

Forsyth, I. (1996). *Teaching and learning materials and the Internet.* London: Kogan Page Limited.

Franklin, S., & Peat, M. (2000, December). *Managing change: The use of mixed delivery modes to increase learning opportunities.* Paper presented at the ASCILITE 2000, Southern Cross University, NSW, Australia.

Gagnon, G. W., & Collay, M. (n.d.). *Constructivist learning design.* Retrieved August 29, 2000, from http://www.prainbow.com/cld/cldp.html

Gokhale, A. A. (1995). Collaborative learning enhances critical thinking. *Technology Education, 7*(1), 22-30.

Goodrich, M. T., & Tamassia, R. (1998, February). *Teaching the analysis of algorithms with visual proofs.* Paper presented at the Proceeding of the Twenty-ninth SIGSCE 98 Technical Symposium on Computer Science Education, Atlanta, Georgia, USA.

Gray, A. (1997). *Constructivist teaching and learning.* Retrieved September 5, 2000, from http://www.ssta.sk.ca/research/instruction/97-07.htm

Gray, J., Boyle, T., & Smith, C. (1998, August). *A constructivist learning environment implemented in Java.* Paper presented at the Proceedings of the 6th Annual Conference on the Teaching of Computing/3rd Annual Conference on Integrating Technology into Computer Science Education on Changing the Delivery of Computer Science Education, Dublin, Ireland.

Gutwill-Wise, J. P. (2001). The impact of active and context-based learning in introductory chemistry courses: An early evaluation of the modular approach. *Journal of Chemical Education, 78*(5), 684-690.

Haddon, K., Smith, C., Brattan, D., & Smith, T. (1995, December). *Can learning via multimedia benefit weaker students?* Retrieved May 10, 2002, from http://www.ilt.ac.uk/public/cti/ActiveLearning/issue3/smith/index.html

Hagan, D., & Lowder, J. (1996, December). *Use of the World Wide Web in introductory computer programming.* Paper presented at the ASCILITE 1996, Adelaide, South Australia.

Hagan, D., & Sheard, J. (1998, August). *The value of discussion classes for teaching introductory programming.* Paper presented at the ITiCSE'98, Dublin, Ireland.

Hansen, S. R., Narayanan, N. H., & Schrimpsher, D. (2000). *Helping learners visualize and comprehend algorithm.* Retrieved May 14, 2003, from http://imej.wfu.edu/articles/2000/1/02/index.asp

Herrington, J. A. (1997). *Authentic learning in interactive multimedia environments.* Unpublished doctoral dissertation, Edith Cowan University, Perth, Western Australia.

Herrmann, N., & Popyack, J. L. (1994, March). *An integrated, software-based approach to teaching introductory computer programming.* Paper presented at the Twenty-fifth Annual SIGCSE: Symposium on Computer Science Education, Phoenix, Arizona, USA.

Higgins, P. (1996). The World Wide Web: An introduction. *Information Development, 12*(3), 145-148.

Holzer, S. M. (1994). *From constructivism......to active learning.* Retrieved September 1, 2000, from http://www.succeed.ufl.edu/pubs/innovator/innovator 2/innovator002.html

Honebein, P. C. (1996). Seven goals for the design of constructivist learning environment. In B. G. Wilson (Ed.), *Constructivist learning environments: Case studies in instructional design* (pp. 11-24). Englewood Cliffs, New Jersey: Educational Technology Publications, Inc.

Hsi, S. (1997). *Facilitating knowledge integration in science through electronic discussion: The multimedia forum kiosk.* Retrieved September 6, 2000, from http://best.me.berkeley.edu/~hsi/dissertation/hsidissy.html#ABS

Hubbard, D. (1996). *Education of computer programming.* Retrieved August 23, 2000, from http://www.gl.umbc.edu/~schmitt/331S96/dhubba1/report.html

Itoh, Y., Konishi, T., & Suzuki, H. (1998, October). *On experimental evaluation of an educational system that supports teachers of novice programming.* Paper presented at the Proceeding of ICCE'98: Global Education on the Net, Beijing, China.

Jehng, J. J., & Chan, T. (1998). Design computer support for collaborative visual learning in the domain of computer programming. *Computer in Human Behavior, 14*(3), 429-448.

Jenkins, T. (1998, August). *A participative approach to teaching programming.* Paper presented at the ITiCSE'98, Dublin, Ireland.

Jenkins, T., & Towle, W. (1997, August). *Teaching programming to novices - Can technology help?* Paper presented at the 5th Annual Conference on the Teaching of Computing, Dublin, Ireland.

Johnson, L. F. (1995). Technical opinion: C in the first course considered harmful. *Communications of the ACM, 38*(5), 99-101.

Johnson-Laird, P. N. (1983). *Mental models: Toward a cognitive science of language, inference, and consciousness.* Cambridge: Cambridge University Press.

Jonassen, D. H. (1991). Objectivism versus constructivism: Do we need a new philosophical paradigm? *Educational Technology Research and Development, 39*(3), 5-14.

Juliussen, E. (2002). *Internet users will top 1 billion in 2005.* Retrieved May 19, 2002, from http://www.c-i-a.com/pr032102.htm

Kann, C., Lindeman, R. W., & Heller, R. (1997). Integrating algorithm animation into a learning environment. *Computers and Education, 28*(4), 223-228.

Karsten, R., & Kaparthi, S. (1998). Using dynamic explanation to enhance novice programming instruction via the World Wide Web. *Computer and Education, 30* (3/4), 195-201.

Karuppan, C. M. (2001). Web-based teaching materials: A user's profile. *Internet Research, 11*(2), 138-148.

Kerlin, B. A., Kerlin, S. P., & Obrien, A. E. (1997). *Teach 2000: The educators' guide to the Internet and World Wide Web*. Retrieved September 14, 2002, from http://www.lhbe.edu.on.ca/teach2000/onramp/why.html

Khan, B. H. (1997). Web-based instruction (WBI): What is it and why is it? In B. H. Khan (Ed.), *Web-based instruction* (pp. 5-18). Englewood Cliffs, New Jersey: Educational Technology Publications, Inc.

Kulik, C.-L. C., & Kulik, J. A. (1986). Effectiveness of computer-based instruction in colleges. *Association for Educational Data Systems Journal, 19*(2/3), 81-108.

Kulik, C.-L. C., & Kulik, J. A. (1991). Effectiveness of computer-based instruction: An updated analysis. *Computer and Human Behaviour, 7*(1/2), 75-94.

Lebow, D. (1993). Constructivist value for instructional systems design: Five principles toward a new mindset. *Educational Technology, Research and Development, 41* (3), 4-16.

Lemay, L. (1996). *Teach yourself Web publishing with HTML 3.2 in 14 days* (Professional Reference ed.). Indianapolis, Indiana: Sam.net Publishing.

Lischner, R. (2000). *Programming language and tools for deep learning*. Retrieved August 22, 2000, from http://www.cs.utexas.edu/users/csed/doc consortium/DC99/lischner-abstract.html

Lorsbach, A., & Tobin, K. (1992). *Constructivism as a referent for science teaching*. Retrieved September 1, 2000, from http://www.exploratorium.edu/IFI/resources/research/constructivism.html

Love, P. A., & Gosper, M. V. (1995, December). *Developing interactive course materials: Using HTML to integrate conventional and Internet resources*. Paper presented at the Twelfth Annual Conference of the Australian Society for Computers in Learning in Tertiary Education: ASCILITE'95, University of Melbourne, Victoria, Australia.

MacDonald, C. J., Stodel, E. J., Farres, L. G., Breithaupt, K., & Gabriel, M. A. (2001). The demand-driven learning model: A framework for Web-based learning. *The Internet and Higher Education, 4*, 9-30.

Madhumita, & Kumar, K. L. (1995). Twenty-one guidelines for effective instructional design. *Educational Technology, 35*(3), 58-61.

Maltby, J. R., & Whittle, J. (2000, December). *Learning programming online: Student perceptions and performance.* Paper presented at the ASCILITE 2000, Southern Cross University, NSW, Australia.

Marshall, C., & Rossman, G. B. (1999). *Designing qualitative research* (3rd ed.). Thousand Oaks, California: Sage Publications, Inc.

Mayer, R. E. (1976). Some conditions of meaningful learning for computer programming: Advance organizers and subject control of frame order. *Journal of Educational Psychology, 68*(2), 143-150.

Mayer, R. E., & Fay, A. L. (1987). A chain of cognitive changes with learning to program in Logo. *Journal of Educational Psychology, 79*(3), 269-279.

McIntyre, M. (1997, December). *The Web as an effective tool for adult learners.* Retrieved April 2, 2002, from http://www.tcm.com/trdev/mcintyre.htm

McKeown, J., & Farrell, T. (2000). *Why we need to develop success in introductory programming courses.* Retrieved August 8, 2000, from http://homepages.dsu.edu/mckeownj/CPCCCSCpaper.html

McKinney, K. (1996). *Example of active learning techniques.* Retrieved September 15, 2000, from http://www.ilstu.edu/depts/CAT/actlearn.htm

McLoughlin, C. (1997, December). *Visual thinking and telepedagogy.* Paper presented at the Australian Society for Computers in Learning in Tertiary Education 97, Perth, Western Australia.

Merriam, S. B. (1998). *Qualitative research and case study applications in education: Revised and expanded from case study research in education.* San Francisco: Jossey-Bass Publishers.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook* (2nd ed.). Thousand Oaks, California: Sage Publications, Inc.

Moore, N. (2000). *How to do research* (3rd ed.). London: Library Association Publishing.

Mulholland, P., & Eisenstadt, M. (Eds.). (1998). *Using software to teach computer programming: Past, present and future.* Cambridge, Massachusetts: MIT Press.

Najjar, L. J. (1996). Multimedia information and learning. *Journal of Educational Multimedia and Hypermedia, 5*(2), 129-150.

Naps, T. L., & Bressler, E. (1998, February). *A multi-windows environment for simultaneous visualization of related algorithms on the World Wide Web.* Paper presented at the Proceeding of the Twenty-ninth SIGSCE 98 Technical Symposium on Computer Science Education, Atlanta, Georgia, USA.

Newby, T. J., Stepich, D. A., Lehman, J. D., & Russell, J. D. (1996). *Instructional technology for teaching and learning: Designing instruction, integrating computers, and using media.* Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

Newlands, D. A. (1992, December). *C as a first programming language.* Paper presented at the Australian Society for Computers in Learning in Tertiary Education, Sydney, Australia.

Norman, D., & Sphorer, J. C. (1996). Learning-centered education. *Communication of the ACM, 39*(4), 24-27.

Oliver, R. (1998). *Teaching and learning on the World Wide Web: A guide for teachers.* Perth, Western Australia: Edith Cowan University.

Oliver, R. (1999a). *On-line teaching and learning: New roles for participants.* Retrieved September 13, 2000, from http://www.monash.edu.au/groups/flt/1999/online.html

Oliver, R. (1999b, October). *Teaching and learning with technology: Learning from experience.* Paper presented at the Proceedings of the Australian Curriculum Assessment and Certification Authorities Conference, Perth. Western Australia.

Oliver, R. (2000, December). *When teaching meets learning: Design principles and strategies for Web-based learning environments that support knowledge construction.* Paper presented at the ASCILITE 2000, Southern Cross University, NSW, Australia.

Oliver, R., & Herrington, J. (2001). *Teaching and learning online: A beginner's guide to e-learning and e-teaching in higher education.* Mt. Lawley, Perth: Centre for Research in Information Technology and Communications, Edith Cowan University.

Oliver, R., Herrington, J., & Omari, A. (1996, July). *Creating effective instructional materials for the World Wide Web.* Paper presented at the AusWeb96: The Second Australian WorldWideWeb Conference, Conrad Jupiters Hotel, Gold Coast, Queensland, Australia.

Oliver, R., & Malone, J. (1993). The influence of instruction and activity on the development of semantic programming knowledge. *Journal of Research on Computing in Education, 25*(4), 521-533.

Oliver, R., Omari, A., & Herrington, J. (1998, December). *Developing converged learning environments for on and off-campus students using the World Wide Web.* Paper presented at the ASCILITE'98, Wollongong, Australia.

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer.* New York: BasicBooks Publishers, Inc.

Patton, M. Q. (1990). *Qualitative evaluation and research methods* (2nd ed.). Newbury Park, California: Sage Publications, Inc.

Perkins, D. N., Hancock, C., Hobbs, R., Martin, F., & Simmons, R. (1986). Condition of learning in novice programmers. *Journal of Education Computing Research, 2*(1), 37-55.

Perkins, D. N., Schwartz, S., & Simmons, R. (1988). Instructional strategies for the problems of novice programmers. In R. E. Mayer (Ed.), *Teaching and learning computer programming: Multiple research perspectives* (pp. 153-178). New Jersey: Lawrence Erlbaum Associated, Inc.

Perry, T., & Perry, L. A. (1998). University students' attitudes towards multimedia presentations. *British Journal of Educational Technology, 29*(4), 375-377.

Pierson, W. C., & Rodger, S. H. (1998, February). *Web-based animation of data structure using JAWAA.* Paper presented at the Proceeding of the Twenty-ninth SIGSCE 98 Technical Symposium on Computer Science Education, Atlanta, Georgia, USA.

Powerpoint in the classroom: Teacher's guide. (1998). Retrieved September 20, 2002, from http://www.actden.com/pp/guide.htm

Race, P. (1996). *A fresh look at independent learning.* Retrieved August 29, 2000, from http://www.lgu.ac.uk/deliberations/eff.learning/indep.html

Rais-Rohani, M. (2001). On effective of an online tutorial to enhance statics instruction. *Computer in Education Journal, 11*(2), 38-43.

Reid, I. C. (2001). Reflection on using the Internet for the evaluation of course delivery. *The Internet and Higher Education, 4*, 61-75.

Relan, A., & Gillani, B. B. (1997). Web-based instruction and the traditional classroom: Similarities and differences. In B. H. Khan (Ed.), *Web-based instruction* (pp. 41-46). Englewood Cliffs, New Jersey: Educational Technology Publications, Inc.

Rieber, L. P. (1995). A historical review of visualization in human cognition. *Educational Technology Research and Development, 43*(1), 45-56.

Ritchey, T. (1996). *Programming JavaScript for Netscape 2.0.* Indianapolis, Indiana: New Riders Publishing.

Ritchie, D. C., & Hoffman, B. (1997). Incorporating instructional design principles with the World Wide Web. In B. H. Khan (Ed.), *Web-based instruction* (pp. 135-138). Englewood Cliffs, New Jersey: Educational Technology Publications, Inc.

Rose, D. H., & Meyer, A. (2002). *Teaching every student in the digital age: Universal design for learning.* Alexandria, Verginia: Association for Supervision & Curriculum Development.

Rowe, G., & Thorburn, G. (1999, August). *Evaluate of VINCE - a tool for teaching introductory programming.* Paper presented at the 7th Annual Conference on the Teaching of Computing, University of Ulster, Northern Ireland.

Sale, J. E. M., Lohfeld, L. H., & Brazil, K. (2002). Revisiting the quantitative-qualitative debate: Implications for mixed-methods research. *Quality & Quantity, 36*, 43-53.

Sangster, A. (1995, July). *World Wide Web - what can it do for education?* Retrieved September 10, 2002, from http://www.ilt.ac.uk/public/cti/ActiveLearning/issue2/sangster/index.html

Savery, J. R., & Duffy, T. M. (1995). Problem based learning: An instructional model and its constructivist framework. *Education Technology, 35*(5), 31-38.

Schank, R. C., & Cleary, C. (1995). *Engines for education.* Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.

Schemmel, J. J., Hall, K. D., & Dennis, N. D. (1997). Using computer technology to enhance instruction and learning. *Computers in Education Journal, 7*(3), 31-36.

Segal, J., & Ahmad, K. (1993). The role of examples in the teaching of programming languages. *Journal of Education Computing Research, 9*(1), 115-129.

Selby, L., & Young, A. (1998, August). *The teaching of programming: Changing the delivery to meet all students' needs.* Paper presented at the Proceedings of the 6th Annual Conference on the Teaching of Computing/3rd Annual Conference on Integrating Technology into Computer Science Education on Changing the Delivery of Computer Science Education, Dublin, Ireland.

Sewell, M. (n.d.). *The use of qualitative interview in evaluation.* Retrieved April 7, 2002, from http://ag.arizona.edu/fcr/fs/cyfar/Intervu5.htm

Shih, Y., & Alessi, M. S. (1993-1994). Mental models and transfer of learning in computer programming. *Journal of Research on Computing in Education, 26*(2), 154-174.

Shim, J. (1998). *Constructing approaches to technology integration into sciences classrooms.* Retrieved August 29, 2000, from http://www.sd38.bc.ca/schools/Palmer/TESSI/Reports/construc.htm

Smith, C. D., Cypher, A., & Tesler, L. (2000). Novice programming comes to age. *Communication of the ACM, 43*(3), 75-81.

Smith, P., & Webb, G. (1998, October). *Overview of a low-level programming visualisation tool for novice C programmers.* Paper presented at the ICCE'98: Global Education on the Net, Beijing, China.

Snyder, I. (1995). Multiple perspectives in literacy research: Integrating the quantitative and qualitative. *Language and Education, 9*(1), 45-49.

Soloway, E. (1986, September). Learning to program = learning to construct mechanisms and explanations. *Communications of the ACM, 29,* 850-858.

Soloway, E., Ehrlich, K., Bonar, J., & Greenspan, J. (Eds.). (1981). *What do novices know about programming?* New Jersey: ABLEX Publishing Corporation.

Spiro, R. J., Feltovich, P. J., Jacobson, M. J., & Coulson, R. L. (1991). Cognitive flexibility, constructivism, and hypertext: Random access instruction for

advanced knowledge acquisition in ill-structured domains. *Educational Technology, 31*(5), 24-33.

Sproull, N. L. (1988). *Handbook of Research Methods: A guide for practitioners and students in the social sciences.* New Jersey: Scarecrow Press, Inc.

Steinberg, E. R. (1991). *Computer-assisted instruction.* Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.

Stephens, J. (n.d.). *Constructivism and science teaching.* Retrieved August 31, 2000, from http://www.bamaed.ua.edu/~wdwyer/CurricSem/UpdateReports/JoanneUpdate.html

Swigger, K. M., Brazile, R., & Shin, D. (1995). *Teaching computer science students how to work together.* Retrieved September 13, 2000, from http://www-cscl95.indiana.edu/cscl95/swigger.html

Tolhurst, D. (1995). Hypertext, Hypermedia, Multimedia defined? *Educational Technology, 35*(2), 21-26.

Treichler, D. G. (1967). Are you missing the boat in training aid? *Film and A-V Communication, 1*, 14-16.

Trentin, G. (1999). What does 'Using the Internet for Education' mean? *Educational Technology, 39*(4), 15-23.

Tripathi, A. K. (1999). *Implication of Internet learning.* Retrieved April 2, 2002, from http://www.edfac.unimelb.edu.au/online-ed/mailouts/1999/may23.html

Trochim, W. M. (2000, August 2). *The research methods knowledge base.* Retrieved April 29, 2002, from http://trochim.human.cornell.edu/kb/quesresp.htm

Turney, B., & Robb, G. (1971). *Research education: An introduction.* Hinsdale, Illinois: The Dryden Press, Inc.

Twigg, C. A. (1993). Can education be "productive"? *Educom Review, 28*(6), 10-12.

Ulrich, K. (2001). *Flash 5: Visual quick start guide for Windows and Macintosh.* Berkeley, California: Peachpit Press.

Vosniadou, S. (1994). From cognitive theory to educational technology. In S. Vosniadou, R. D. Corte & H. Mandl (Eds.), *Technology-based learning environments: Psychological and educational foundations* (pp. 11-18). Germany: Springer-Verlag.

Wahle, C. P. (1998, February 11, 2001). *A guide to multimedia in education.* Retrieved September 16, 2002, from http://wonders.eburg.wednet.edu/Topics/GMMIE/

Waller, R. (n.d., October 16, 2001). *How big is the Internet.* Retrieved May 10, 2002, from http://www.waller.co.uk/web.htm

Warendorf, K. (1997, September). *ADIS - an animated data structure intelligent tutoring system on the World Wide Web.* Paper presented at the International Conference on Information, Communications and Signal Processing ICICS'97, Singapore.

Watson, G. (2002). *MP3.* Retrieved March 14, 2002, from http://256.com/cgw/service/mp3.html

Wave File. (2002). Retrieved March 14, 2002, from http://whatis.techtarget.com/definition/0,,sid9_gci213473,00.html

WebSideStory. (2001). *Macromedia's Flash doubles market penetration on Netscape browsers.* Retrieved September 19, 2002, from http://www.websidestory.com/cgi-bin/wss.cgi?corporate&news&press_1_127

Whalley, W. B. (1995, July). *Using the Internet for teaching: Teaching and learning on the Internet.* Retrieved May 10, 2002, from http://www.ilt.ac.uk/public/cti/ActiveLearning/issue2/whalley/index.html

Wilcocks, D., & Sanders, I. (1994). Animating recursion as an aid to instruction. *Computer and Education, 23*(3), 221-226.

Yang, T.-J. (1998). *Teaching C programming on the World Wide Web: A system and evaluation.* Unpublished doctoral dissertation, The Graduate College at the University of Nebraska, Lincoln, Nebraska, USA.

Yoo, Y. D. (1998, October). *An interactive hypermedia learning system: LMM.* Paper presented at the ICCE'98: Global Education on the Net, Beijing, China.

Young, M. F. (1993). Instructional design for situated learning. *Educational Technology Research and Development, 41*(1), 43-58.

Zakari, M. (1998). *Constructivism: Is it the last paradigm shift "war?"* Retrieved August 31, 2000, from http://www2.educ.ksu.edu/Faculty/McGrathD/Spring98/StudentCogProjects/constructivism.htm

APPENDICES

Appendix 1: Ethics Clearance Forms

- Letter of Information
- Consent Form

Appendix 2: Pilot Study

- Self-administered Evaluation Questionnaire Form

Appendix 3: Research Forms

- Tutor Observation Form
- Researcher Observation Form
- Subject Semi-structured Interview Form
- Self-administered Evaluation Questionnaire Form
- Tutor Semi-structured Interview Form
- Weekly Task

APPENDIX 1: ETHICS CLEARANCE FORMS

- **Letter of Information**
- **Consent Form**

Letter of Information

(To be sent out if possible by Rector of Suranaree University of Technology)

Dear students,

We have been approached by Mr. Kacha Chansilp, who is one of our lecturers and now studying Ph.D. in Edith Cowan University, Perth, Western Australia, to conduct a research at our institute. He is hoping to find out about how students learn in introductory computer programming courses.

This research is important because it will help us to know students' performance. This will eventually help us to improve our curriculum, materials of introductory computer programming courses and teaching methodology. The more we can understand these things the more we can help students through our teaching.

The research will centre on the use of computer-based learning aids. The research will also give each student a chance to test his/her ability in learning how to program and to share his/her ideas and perception.

You are selected to participate in this research and we hope that you will be cooperative. In the study all you have to do is attending in the laboratory. You will be given a short weekly task to complete and check your understanding via the use of a Web-based instructional tool. Three voluntary students will be asked for an interview and a screen video capture software will be use to record the use of a Web-based instructional tool. If for some reason you feel uncomfortable, you can stop immediately.

The data collected will be treated with the strictest confidence. You will not be identified by name in any reports of this research.
If you are pleased to participate in this research, could you please complete the form below and return it to the institute. If you do not wish to participate, we respect your right. If this is the case do not return the form to the institute. You needn't do anything. Your decision will not result in any loss of benefits to which you are otherwise entitled.
If you have any queries about this research, please contact Mr. Kacha Chansilp in the School of Computer Engineering, Institute of Engineering, 111 University Avenue, Muang District, Nakhon Ratchasima 30000. Phone [redacted]

Thank you.
Yours Sincerely,

(Asst. Prof. Dr. Tavee Lertpanyavit)
Rector of Suranaree University of Technology

Development, Implementation and Evaluation of an Interactive Multimedia Instructional Model: A Teaching and Learning Programming Approach

Consent Form

I am satisfied with the information provided to me on the computer programming research project to be conducted by Mr. Kacha Chansilp, a Ph.D. student of Edith Cowan University and SUT lecturer.

I'm happy to take part in the project. I understand that no personal or identifiable reference will be made without my permission in any publication connected with the project.

Name: ______________________________ Date: ______________

Signature: ____________________________

APPENDIX 2: PILOT STUDY

- **Self-administered Evaluation Questionnaire Form**

DIVTIC2001: Pilot Study

Hi colleague,

Please welcome to a pilot study of a dynamic interactive visualisation tool for C (DIVTIC.) I would like you to proceed and analyse the Animated Examples in DIVTIC system and give me the feedback.

Animated Example Menu contains 46 animations that covers the following topics:

Group A

Flowchart (5 animations)
- 1. Flowchart Symbols
- 1a. Credit Approval
- 1b. Circle in the Square
- 1c. Menu Display
- 1d. Samples

Data Types & Input/Output (5 animations)
- 2. Data Basics
- 2a. Input/Output *using scanf( ) and printf( )*
- 2b. Input/Output *using getchar( ), putchar( ), gets( ), and puts( )*
- 2c. Area of Triangle
- 2d. Triangular Circumferences

Operators (5 animations)
- 3. Operators
- 3a. Basic Arithmetic Operator
- 3b. Condition Operator
- 3c. Increment/Decrement Operator
- 3d. Dividing a Number

Control (4 animations)
- 4. General control Forms
- 4a. Print value of I from 1 to 3
- 4b. Averaging numbers

Group B

- 4c. Hollow Triangle

Functions (4 animations)
- 5. Functions and Library
- 5a. Lucky Number
- 5b. Double Character
- 5c. Reverse String

Arrays (3 animations)
- 6. Arrays?
- 6a. Array Summation
- 6b. Vowel Counter

Pointers (5 animations)
- 7. Pointer Bascis
- 7a. Assigning and Printing Pointers
- 7b. Averaging an Array Using Pointer
- 7c. Pointer Array

Sorting & Searching (5 animations)
- 8. Sorting & Searching Overview
- 8a. Numeric Sorting
- 8b. String Sorting
- 8c. Linear Search

Group C

8d. Binary Search

Structure (4 animations)

9. Structure Basics
9a. Creating a Basic Structure
9b. Pointer and an Array of Structure
9c. Pointer, Function, and Structure

Data File (6 animations)

10. Data File?
10a. Writing a Simple Text File
10b. Reading a Text File
10c. Appending a Text File
10d. Writing a Simple Binary File
10e. Reading a Binary File

Please take your time and analyze the animation in order from the beginning to the end, Topic 1 to 10, e.g., 1 ➔1a ➔ 1b ➔ ... ➔ 5a ➔ 5b ➔ 5c ➔ ...10e.

You have selected to be one of these three groups, Group A, Group B, or Group C. So, you please answer all the questions in the supplied Q/A sheet for related animations (one sheet for one animation) based on your comprehension and analysis. Your cooperation would be truly grateful and you would be a part of a group who help developing in an educational technology tool in teaching and learning introductory computer programming course such as DIVTIC.

Again, thank you very much for your time and cooperation.

Researcher: Kacha Chansilp
E-mail: [redacted]

NOTE: To run the animation:
Group A: Double click "Divtic_A.exe",
Group B: Double click "Divtic_B.exe", or
Group C: Double click "Divtic_C.exe".

Then click on the "Animated Examples" button and rollover the topic you need to access, the subheadings will appear on the right side, then you can simply select the animation you want to analyse.

DIVTIC2001: Pilot Study

Animation Title: ____________________ **Date:** __________

Code: __________

Part 1: Please indicate how much you disagree or agree with each of the following statements.

Note: 1 = Strongly Disagree
2 = Disagree
3 = Not Applicable
4 = Agree
5 = Strongly Agree

Question	Strongly Disagree 1	Disagree 2	Not Applicable 3	Agree 4	Strongly Agree 5
The speed of the animation is okay.					
The animation is easy to understand in concept.					
The animation process is easy to control.					
The interface of the animation is good.					
The animation is easy to navigate.					
The message board is well explained in details.					
The memory map is well explained.					
The problem is too difficult.					

Part 2: Please fill in the following questions

1. Is the animation accurate in all respects? Yes No
   If No, please explain

2. Please suggest any improvements that will make this a better example

APPENDIX 3: RESEARCH FORMS

- Tutor Observation Form
- Researcher Observation Form
- Subject Semi-structured Interview Form
- Self-administered Evaluation Questionnaire Form
- Tutor Semi-structured Interview Form
- Weekly Task

Tutor Observation Form

DIVTIC 2001

Name:____________________________ **Date:**______________

Please answer the following questions.

Question 1: What questions did student ask when using DIVTIC?

Question 2: How easy was DIVTIC for students?

Question 3: What problems did you face in this week relating to DIVTIC?

Hardware:

Software:

Network:

Others:

THANK YOU FOR YOUR COOPERATION

Researcher Observation Form

DIVTIC 2001

Researcher:____________________ **Date:**__________

Problems in this week?

Hardware:

Software:

Network:

Others:

Self-administered Evaluation Questionnaire Form

DIVTIC 2001

Student Name: ______________________ **ID:** ____________ **Date:** ________

Part 1: Please indicate how much you disagree or agree with each of the following statements.

Note: 1 = Strongly Disagree
2 = Disagree
3 = Not Applicable
4 = Agree
5 = Strongly Agree

Question	1 Strongly Disagree	2 Disagree	3 Not Applicable	4 Agree	5 Strongly Agree
When I am watching the animation, I stop it from time to time to reflect on what I am trying to get out of it.					
Using Animated Examples help me to think logically during the animation process.					
I always discuss with my peers about the animation running process.					
Using Animated Examples increase my confidence in learning programming.					
Using Animated Examples, I believe that I can solve more complicated tasks.					
Using Animated Examples, I feel that I can help other peers in solving a given problem.					
Using Animated Examples, I feel that I pay more attention in programming class.					
Using Animated Examples encourage me in programming more efficiently.					
Using Animated Examples, I feel that programming is not too difficult to learn.					
The interface of Animated Examples is pleasant.					
Animated Examples are an easy-to-use tool.					
Animated Examples are easy to navigate.					
I enjoy using Animated Examples.					
Animated Examples entertain me in learning programming.					
I feel comfortable by using Animated Examples.					
I am pleased to have Animated Examples as assistance.					
Animated Examples are a useful tool in learning how to program in C.					
Animated Examples material is challenging.					
Using Animated Examples are boring.					
Animated Examples cause mental weariness.					
I feel asleep when I use Animated Examples.					
I can apply the concept getting from Animated Examples to solve given problems.					
Using Animated Examples enhance my understanding.					
Animated Examples cover all I need to learn in programming appropriately.					
Concepts are addressed and well explained in Animated Examples.					
Animated Examples clarify me in learning the programming environment.					
Using Animated Examples, I comprehend how the program executes.					
I always discuss programming problems with my peers.					
I always post the programming problems to the WebBoard.					
It is more effective to learn Animated Examples in a group rather than individually.					
I have used a computer for many years.					
I am an experienced Internet user.					
I have no difficulty in using computer.					

Please turn over for Part 2

Part 2: Please ticks all that apply.

1. How many times have you used the following parts of DIVTIC in the last two weeks?

Topic	0	1 - 3	4 - 6	7 - 10	> 10
Syllabus/Lecture Note					
Computer Structure					
Animated Examples					
"C" Compiler					
"C" WebBoard					
Self-evaluation					
FAQ Pool					
"C" References & Links					

2. How do you use "Animated Example"?

- ☐ I just watch the animation without interacting with it.
- ☐ I press **STOP/PLAY** button to think about and change what is going on.
- ☐ I repeat the animation to make it clear how the program executes.
- ☐ I press Backward/Forward to see the animation.
- ☐ I stop the animation and discuss with my peers.
- ☐ I go to test my own code right after watching the animation to compare the output.
- ☐ Others:____________________

3. When do you normally use DIVTIC?

- ☐ When I am in a laboratory session.
- ☐ When I have free time.
- ☐ When I do assignments.
- ☐ Before examinations.
- ☐ When I face with programming problems.
- ☐ When I am with friends.
- ☐ Others:____________________

Part 3: Please describes the following factors:

What problems in relation to DIVTIC did you face in last two weeks?

☐ Hardware:____

____

____

☐ Software: ____

____

____

☐ Network: ____

____

____

☐ Others: ____

____

____

Tutor Semi-structured Interview Form

DIVTIC 2001

Interviewer:________________________________ **Date:**____________

Interviewee:____

Question 1: What do you think about **Animated Examples** e.g., interface, useability, clarity, user-friendliness, and value?

____
____
____
____
____

Question 2: As a tutor/instructor, do you like **Animated Examples**, how and why?

____
____
____
____
____

Question 3: As a student, do you like **Animated Examples**, how and why?

____
____
____
____
____

Question 4: What other features do you think **Animated Examples** should have?

____
____
____
____
____

Question 5: Do you have any other comments about **Animated Examples**?

____
____
____
____
____

DIVTIC Weekly Task – Week 1: Flowchart

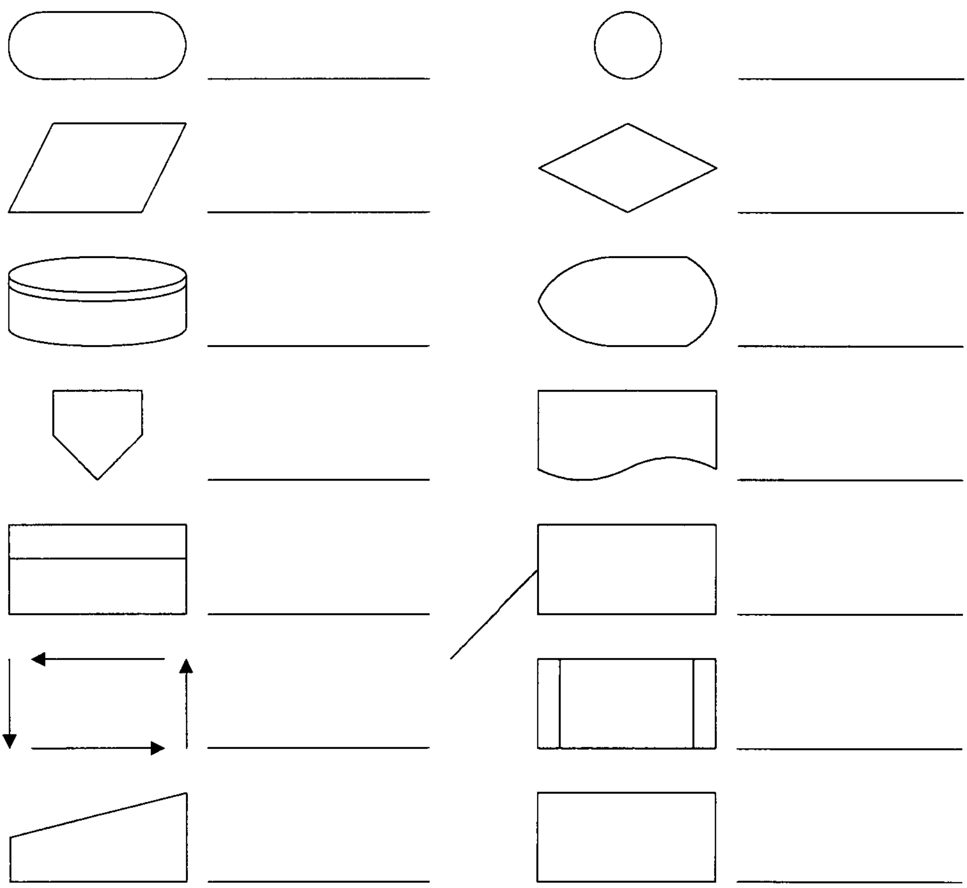
/10

Name: ____________________

Student ID: ____________________ **Date:** ____________

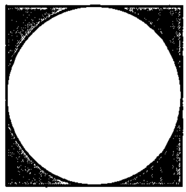
Week 1: Flowchart

Problem 1: Please identifies the following symbols:



Note: Compare to *"Flowchart: Flowchart Symbols"*
Why I was wrong?

Problem 2: User will enter a radius, then draw a flowchart to calculate the area of non-overlap square (grey area) and print out the result to screen.



Example of a running program

Please enter radius: *3*

The area of non-overlap square: *7.725691*

Note: Compare to *"Flowchart: Circle in the Square"*

Why I was wrong?

DIVTIC Weekly Task – Week 2: Data Types & I/O

Name: ______________________________

Student ID: ____________________ **Date:** ____________

/10

Week 2: Data Types & I/O

Problem 1: From a given code below, please answer the following questions:

```
#include<stdio.h>                              /* line 1 */
main( ){                                       /* line 2 */
   char  n[16], c;                             /* line 3 */
   int  age;                                   /* line 4 */
   float  gpa;                                 /* line 5 */
   printf("Name : ");                          /* line 6 */
   scanf(" %s",n);                             /* line 7 */
   printf("Age: ");                            /* line 8 */
   scanf(" %d",&age);                          /* line 9 */
   printf("GPA: ");                            /* line 10 */
   scanf(" %f",&gpa);                          /* line 11 */
   printf("Expected Grade : ");                /* line 12 */
   scanf(" %c",&c);                            /* line 13 */
   printf("Hi... %s,  ",n);                    /* line 14 */
   printf("you are  %d  years old.\n",age);    /* line 15 */
   printf("Your GPA is %f  , ",gpa);           /* line 16 */
   printf("and you have expected  %c.\n",c);   /* line 17 */
   return 0;                                   /* line 18 */
}                                              /* line 19 */
```

1

Your Answer

Q1: How many byte does this program need to use in memory?
A1: ____________________

Q2: If the user enters ***"Somsak"*** in line 6, what will happen in the memory when using ***scanf( )*** in line 7?
A2: ____________________

2

DIVTIC System

Q1: How many byte does this program need to use in memory?
A1: ____________________

Q2: If the user enters ***"Somsak"*** in line 6, what will happen in the memory when using ***scanf( )*** in line 7?
A2: ____________________

3

Note: Compare it to *"Data Types: Input/Output using scanf( ) and printf( )"*

Why I was wrong?

____

____

____

____

Problem 2: From a given code below, please answer the following questions:

```
#include<stdio.h>                                  /* line 1 */
main( ){                                           /* line 2 */
                                                   /* line 3 */
        float  base, height, area;                 /* line 4 */
        printf("Base: ");                          /* line 5 */
        scanf("%f",&base);                         /* line 6 */
        printf("Height: ");                        /* line 7 */
        scanf("%f", &height);                      /* line 8 */
        area = (base * height) / 2;                /* line 9 */
        printf("The area is: %.2f",area);          /* line 10 */
                                                   /* line 11 */
        return 0;                                  /* line 12 */
}                                                  /* line 13 */
```

1

Your Answer

Q1: How many bytes do this program need to use in memory?

A1:____________________________

Q2: What output will be produced on the screen in line 10 if the user enters 7 for ***base*** and 5 for ***height***?

A2:____________________________

2

DIVTIC System

Q1: How many bytes do this program need to use in memory?

A1:____________________________

Q2: What output will be produced on the screen in line 10 if the user enters 7 for ***base*** and 5 for ***height***?

A2:____________________________

3

Note: Compare it to *"Data Types: Area of Triangle"*

Why I was wrong?

____

____

____

____

____

____

____

DIVTIC Weekly Task – Week 3: Operators

/10

Name: ____________________

Student ID: ____________________ **Date:** ____________

Week 3: Operators

Problem 1: From a given code below, please answer the following questions:

```
#include <stdio.h>                               /* line 1 */
main( ){                                         /* line 2 */
    int  j = 5,  k = 5;                          /* line 3 */
    printf("j = %d,  k = %d\n",j,k);             /* line 4 */
    printf("j = %d,  k = %d\n",j++, k--);        /* line 5 */
    printf("j = %d,  k = %d\n",j, k);            /* line 6 */
    printf("j = %d,  k = %d\n",++j, --k);        /* line 7 */
    return 0;                                    /* line 8 */
}                                                /* line 9 */
```

1

Your Answer

Q: What is the output after each line?

A:

Line 4: j = ☐ , k = ☐

Line 5: j = ☐ , k = ☐

Line 6: j = ☐ , k = ☐

Line 7: j = ☐ , k = ☐

2

DIVTIC System

Q: What is the output after each line?

A:

Line 4: j = ☐ , k = ☐

Line 5: j = ☐ , k = ☐

Line 6: j = ☐ , k = ☐

Line 7: j = ☐ , k = ☐

3

Note: Compare it to *"Operators: Increment & Decrement Operators"*

Why I was wrong?

____

____

____

____

____

Problem 2: From a given code below, please answer the following questions:

```
#include <stdio.h>                                          /* line 1 */
main( ){                                                    /* line 2 */
                                                            /* line 3 */
  int a, b, c, d, sum=0, num;                               /* line 4 */
  printf("Enter a number : ");                              /* line 5 */
  scanf("%d",&num);                                         /* line 6 */
  a =  num / 1000;                                          /* line 7 */
  b = (num % 1000) / 100;                                   /* line 8 */
  c = (num % 100) / 10;                                     /* line 9 */
  d = num % 10;                                             /* line 10 */
  sum = a + b + c + d;                                      /* line 11 */
  printf("Result of %d + %d + %d + %d = %d\n",a,b,c,d,sum); /* line 12 */
                                                            /* line 13 */
  return 0;                                                 /* line 14 */
}                                                           /* line 15 */
```

1

Your Answer

Q: What is the value of output after each line if user enters 1924?

A:

Line7: a =

Line8: b =

Line9: c =

Line10: d =

Line11: sum =

2

DIVTIC System

Q: What is the value of output after each line if user enters 1924?

A:

Line7: a =

Line8: b =

Line9: c =

Line10: d =

Line11: sum =

3

Note: Compare it to *"Operators: Dividing a Number"*

Why I was wrong?

DIVTIC Weekly Task – Week 4: Control

/10

Name:____

Student ID:______________________ **Date:**____________

Week 4: Control

Problem 1: From a given code below, please answer the following questions:

```
#include <stdio.h>                              /* line 1 */
main( ){                                        /* line 2 */
   int  i;                                      /* line 3 */
   for (i = 1;  i <= 3;  i++)  {                /* line 4 */
      printf("The value of  i is:  ");          /* line 5 */
      printf("%d\n", i);                        /* line 6 */
   }                                            /* line 7 */
   return 0;                                    /* line 8 */
}                                               /* line 9 */
```

1

Your Answer

Q1: When *i* is equal to 2, what output will be produced on the screen?

A1:____________________________

____

Q2: When will this loop stop?

(a). i = 3
(b). i = 4
(c). i = 5
(d). a, b, and c.

2

DIVTIC System

Q1: When *i* is equal to 2, what output will be produced on the screen?

A1:____________________________

____

Q2: When will this loop stop?

(a). i = 3
(b). i = 4
(c). i = 5
(d). a, b, and c.

3

Note: Compare it to *"Control: Print Value of i from 1 to 3"*

Why I was wrong?

____

____

____

____

____

Problem 2: From a given code below, please answer the following questions:

```
#include<stdio.h>                                /* line 1 */
main( ){                                         /* line 2 */
                                                 /* line 3 */
    int  n,  c = 1;                              /* line 4 */
    float  x, avg, sum = 0.0;                    /* line 5 */
    printf("How many number? ");                 /* line 6 */
    scanf("%d",&n);                              /* line 7 */
    while  ( c <= n) {                           /* line 8 */
            printf("x = ");                      /* line 9 */
            scanf("%f",&x);                      /* line 10 */
            sum += x;                            /* line 11 */
            ++c;                                 /* line 12 */
     }                                           /* line 13 */
    avg = sum / n;                               /* line 14 */
    printf("\n The average is %f\n",avg);        /* line 15 */
    return 0;                                    /* line 16 */
}                                                /* line 17 */
```

1

Your Answer

Suppose: n = 3 and user enters the numbers in the ***while*** loop as 4, 7, and 9 respectively.

Q1: What will be the value of ***sum*** when c = 2?

A1: sum = ______

Q2: What will be the value of the following valiables when the loop stops?

A2: sum = ______

avg = ______

2

DIVTIC System

Suppose: n = 3 and user enters the numbers in the ***while*** loop as 4, 7, and 9 respectively.

Q1: What will be the value of ***sum*** when c = 2?

A1: sum = ______

Q2: What will be the value of the following valiables when the loop stops?

A2: sum = ______

avg = ______

3

Note: Compare it to *"Control: Averaging a List of Number"*

Why I was wrong?

____

____

____

____

____

DIVTIC Weekly Task – Week 5: Functions

Name: ______________________________

Student ID: ____________________ **Date:** ____________

/10

Week 5: Functions

Problem 1: From a given code below, please answer the following questions:

```
#include<stdio.h>                                      /* line 1 */
main( ){                                               /* line 2 */
    int  LuckyNumber(int, int, int);                   /* line 3 */
    int  day = 5,  month = 11, year = 2522;            /* line 4 */
    int  n;                                            /* line 5 */
                                                       /* line 6 */
    n = LuckyNumber(day, month, year);                 /* line 7 */
    printf("\nInput: %d-%d-%d\n", day, month, year);   /* line 8 */
    printf("Your lucky number is %d", n);              /* line 9 */
                                                       /* line 10 */
    return 0;                                          /* line 11 */
}                                                      /* line 12 */
                                                       /* line 13 */
int LuckyNumber(int a, int b, int c){                  /* line 14 */
                                                       /* line 15 */
   return ((a+b+c) / a);                               /* line 16 */
}                                                      /* line 17 */
```

1

Your Answer

Q1: What will be the output for line8?

A1: ______________________

Q2: What will be the output for line9?

A2: ______________________

Q3: What is the return value from line16?

A3: ______________________

2

DIVTIC System

Q1: What will be the output for line8?

A1: ______________________

Q2: What will be the output for line9?

A2: ______________________

Q3: What is the return value from line16?

A3: ______________________

3

Note: Compare it to *"Function: Lucky Number"*

Why I was wrong?

____

____

____

____

____

Problem 2: From a given code below, if user enters a string ***"Suranaree"***, please answer the following questions:

```
#include<stdio.h>                                  /* line 1 */
#include<string.h>                                 /* line 2 */
main( ){                                           /* line 3 */
   int backward(char [ ]);                         /* line 4 */
   char str[10];                                   /* line 5 */
   int  n;                                         /* line 6 */
   printf("Please enter a string : ");             /* line 7 */
   gets(str);                                      /* line 8 */
   n = backward(str);                              /* line 9 */
   printf("\nThere are %d characters.", n);        /* line 10 */
   return 0;                                       /* line 11 */
}                                                  /* line 12 */
                                                   /* line 13 */
int backward(char s[ ]){                           /* line 14 */
   int   i, j;                                     /* line 15 */
   j = strlen(s);                                  /* line 16 */
   printf("%s is backward to ",s);                 /* line 17 */
   for(i = j-1; i >= 0; i--)                       /* line 18 */
        printf("%c",s[i]);                         /* line 19 */
   return j;                                       /* line 20 */
}                                                  /* line 21 */
```

1

Your Answer

Q1: Why do we need line 2 ***#include<string.h>?***

A1: ______________________

Q2: What does line 4 do?

A2: ______________________

Q3: What is the value of j in line16?

A3: j = ☐

Q4: What character will be printed in line 19 when i = 6?

A4: Character: ☐

2

DIVTIC System

Q1: Why do we need line 2 ***#include<string.h>?***

A1: ______________________

Q2: What does line 4 do?

A2: ______________________

Q3: What is the value of j in line16?

A3: j = ☐

Q4: What character will be printed in line 19 when i = 6?

A4: Character: ☐

3

Note: Compare it to *"Functions: Reverse String"*

Why I was wrong?

____

____

____

____

____

DIVTIC Weekly Task – Week 6: Arrays

Name: ____________________

Student ID: ____________________ **Date:** ____________

Week 6: Arrays

/ 10

Problem 1: From a given code below, please answer the following questions:

```
#include<stdio.h>                             /* line 1 */
main( ){                                      /* line 2 */
   int n[5] = {7, 12, 5, 20, 9};              /* line 3 */
   int  i,  total = 0;                        /* line 4 */
                                              /* line 5 */
   for(i = 0; i < 5; i++)                     /* line 6 */
       total = total + n[i];                  /* line 7 */
                                              /* line 8 */
   printf("The total is %d\n",total);         /* line 9 */
   if ((total % 2) == 0)                      /* line 10 */
     printf("It is an even number.");         /* line 11 */
   else                                       /* line 12 */
     printf("It is an odd number.");          /* line 13 */
   return 0;                                  /* line 14 */
}                                             /* line 15 */
```

1

Your Answer

Q1: What is the value of ***n[3]*** ?

A1: n[3] = ☐

Q2: What will be the value of ***total*** when i = 3?

A2: total = ☐

Q3: What line will be printed (11or13)?

A3: line = ☐

2

DIVTIC System

Q1: What is the value of ***n[3]*** ?

A1: n[3] = ☐

Q2: What will be the value of ***total*** when i = 3?

A2: total = ☐

Q3: What line will be printed (11or13)?

A3: line = ☐

3

Note: Compare it to *"Array: Array Summation"*

Why I was wrong?

____

____

____

____

____

Problem 2: From a given code below, please answer the following questions:

```
#include<stdio.h>                                                        /* line 1 */
#include<string.h>                                                       /* line 2 */
main( ){                                                                 /* line 3 */
    void VowelCount(char [], int);                                       /* line 4 */
    char str[8];   int l;                                                /* line 5 */
    printf("Please enter a string: ");                                   /* line 6 */
    gets(str);                                                           /* line 7 */
    l = strlen(str);                                                     /* line 8 */
    VowelCount(str, l);                                                  /* line 9 */
    return 0;                                                            /* line 10 */
}                                                                        /* line 11 */
                                                                         /* line 12 */
void VowelCount(char s[ ],   int n){                                     /* line 13 */
    int i,  total = 0;                                                   /* line 14 */
    for(i = 0;  i < n;  i++)                                             /* line 15 */
       if ((s[i] == 'a') || (s[i] == 'e') || (s[i] == 'i') || (s[i] == 'o') || (s[i] == 'u'))   /* line 16 */
              total++;                                                   /* line 17 */
    printf("The word \"%s\" has %d vowel(s).", s, total);                /* line 18 */
    return;                                                              /* line 19 */
}                                                                        /* line 20 */
```

1

Your Answer

Suppose: user enters ***"love"***, then

Q1: What is the value of ***l*** (line 8)?

A1: l = ☐

Q2: What is the value of ***n*** (line 13)?

A2: n = ☐

Q3: What is the value of ***total*** when ***i*** is equal to 1?

A3: total = ☐

2

DIVTIC System

Suppose: user enters ***"love"***, then

Q1: What is the value of ***l*** (line 8)?

A1: l = ☐

Q2: What is the value of ***n*** (line 13)?

A2: n = ☐

Q3: What is the value of ***total*** when ***i*** is equal to 1?

A3: total = ☐

3

Note: Compare it to *"Arrays: Vowel Counter"*

Why I was wrong?

____

____

____

____

____

DIVTIC Weekly Task – Week 7: Pointers

/10

Name:____

Student ID:__________________________ **Date:**____________

Week 7: Pointers

Problem 1: From a given code below, please answer the following questions:

```
#include<stdio.h>                              /* line 1 */
main( ) {                                      /* line 2 */
   int n[5] = {15, 12, 3, 55, 42},   i;        /* line 3 */
   int *ip;                                    /* line 4 */
   float sum= 0.0,  avg=0.0;                   /* line 5 */
                                               /* line 6 */
   ip = n;                                     /* line 7 */
   for(i = 0;  i < 5;  i++)                    /* line 8 */
      sum = sum + *(ip + i);                   /* line 9 */
   avg = sum/5;                                /* line 10 */
   printf("Sum = %f\n", sum);                  /* line 11 */
   printf("Average = %f", avg);                /* line 12 */
                                               /* line 13 */
   return 0;                                   /* line 14 */
}                                              /* line 15 */
```

1

Q1: What is happening in line 7?

A1: Line 7:

Q2: What is the value of ***sum*** when ***i*** is equal to 4?

A2: sum =

Q3: Where does the pointer ***ip*** point to when ***i*** is equal to 3?

A3:________________________

2

DIVTIC System

Q1: What is happening in line 7?

A1: Line 7:____________________

Q2: What is the value of ***sum*** when ***i*** is equal to 4?

A2: sum = [ ]

Q3: Where does the pointer ***ip*** point to when ***i*** is equal to 3?

A3:________________________

3

Note: Compare it to *"Pointers: Averaging an Array using Pointer"*

Why I was wrong?

____

____

____

____

____

Problem 2: From a given code below, please answer the following questions:

```
#include<stdio.h>                                      /* line 1 */
main() {                                               /* line 2 */
        int m = 0, n = 1, k = 2, *ip;                  /* line 3 */
        char msg[] = "hello world", *cp;               /* line 4 */
        printf("Original msg = \"%s\"\n",msg);         /* line 5 */
        ip = &m;                                       /* line 6 */
        *ip = 1;                                       /* line 7 */
        k = *ip;                                       /* line 8 */
        cp = msg;                                      /* line 9 */
        *cp = 'H';                                     /* line 10 */
        cp = &msg[6];                                  /* line 11 */
        *cp = 'W';                                     /* line 12 */
        printf("m = %d, n = %d, k = %d\n",m,n,k);      /* line 13 */
        printf("Updated msg = \"%s\"",msg);            /* line 14 */
                                                       /* line 15 */
        return 0;                                      /* line 16 */
}                                                      /* line 17 */
```

1

Your Answer

Q: What is happening after each line?

A: Line 5: ________________

Line 7: ________________

Line 8: ________________

Line 10: ________________

Line 12: ________________

Line 13: ________________

Line 14: ________________

2

DIVTIC System

Q: What is happening after each line?

A: Line 5: ________________

Line 7: ________________

Line 8: ________________

Line 10: ________________

Line 12: ________________

Line 13: ________________

Line 14: ________________

3

Note: Compare it to *"Pointers: Accessing String using Pointer"*

Why I was wrong?

____

____

____

____

____

DIVTIC Weekly Task – Week 8: Sorting & Searching

/10

Name: ______________________________

Student ID: ____________________ **Date:** ____________

Week 8: Sorting & Searching

Problem 1: From a given code below, please answer the following questions:

```
#include<stdio.h>                                /* line 1 */
#define SIZE   5                                 /* line 2 */
main( ){                                         /* line 3 */
    int n[SIZE]={21,  5,  53,  1,  8};           /* line 4 */
    int  i,  pass,  temp;                        /* line 5 */
    printf("\nBefor Sorting: ");                 /* line 6 */
    for(i = 0; i < SIZE; i++)                    /* line 7 */
        printf("%d  ", n[i]);                    /* line 8 */
    for(pass = 1; pass <= SIZE - 1; pass++)      /* line 9 */
        for(i = 0; i <= SIZE - 2; i++){          /* line 10 */
            if(n[i] > n[i+1]){                   /* line 11 */
                temp = n[i];                     /* line 12 */
                n[i] = n[i+1];                   /* line 13 */
                n[i+1] = temp;                   /* line 14 */
            }                                    /* line 15 */
        }                                        /* line 16 */
    printf("\n\nAfter Sorting:  ");              /* line 17 */
    for(i = 0;  i < SIZE; i++)                   /* line 18 */
        printf("%d  ",n[i]);                     /* line 19 */
    return 0;                                    /* line 20 */
}                                                /* line 21 */
```

1

Your Answer

Q1: When the loop in line 7 will terminate?

A1: ______________________

Q2: What is the value of each member in array ***n,*** just right before ***pass*** is equal to 2?

A2: n[ ] = ☐ ☐ ☐ ☐ ☐

Q3: What is the value of n[1] and n[2] when ***pass = 2*** and ***i = 1***?

A3: n[1] = ☐, n[2] = ☐

2

DIVTIC System

Q1: When the loop in line 7 will terminate?

A1: ______________________

Q2: What is the value of each member in array ***n,*** just right before ***pass*** is equal to 2?

A2: n[ ] = ☐ ☐ ☐ ☐ ☐

Q3: What is the value of n[1] and n[2] when ***pass = 2*** and ***i = 1***?

A3: n[1] = ☐, n[2] = ☐

3

Note: Compare it to *"Sorting & Searching: Numeric Sorting"*

Why I was wrong?

____

____

Problem 2: From a given code below, please answer the following questions:

```
#include<stdio.h>                                   /* line 1 */
main( ){                                            /* line 2 */
    int n[8] = {75, 35, 20, 90, 85, 63, 20, 41};    /* line 3 */
    int i, key, index = -1;                         /* line 4 */
    printf("Enter a number to be searched: ");     /* line 5 */
    scanf("%d",&key);                               /* line 6 */
                                                    /* line 7 */
    for(i = 0; i < 8 ; i++)                         /* line 8 */
      if(key == n[i]) {                             /* line 9 */
        index = i;                                  /* line 10 */
        printf("\nFound at index =   %d",index);    /* line 11 */
      }                                             /* line 12 */
                                                    /* line 13 */
    if(index == -1)                                 /* line 14 */
      printf("\nCannot find key %d", key);          /* line 15 */
                                                    /* line 16 */
    return 0;                                       /* line 17 */
}                                                   /* line 18 */
```

1

Your Answer

Suppose that ***key = 20***,

Q1: When the loop in line 8 will terminate?

A1: ____________________

Q2: What is the value of ***index*** when ***i = 6*** in line 8?

A2: index = ☐

Q3: When line 15 will be executed?

(please tick all that appropriate)

A3: ☐ i = 8; ☐ index = -1;

☐ n[i] = -1; ☐ Never!

☐ Line 14 is TRUE

☐ Line 14 is FALSE

2

DIVTIC System

Suppose that ***key = 20***,

Q1: When the loop in line 8 will terminate?

A1: ____________________

Q2: What is the value of ***index*** when ***i = 6*** in line 8?

A2: index = ☐

Q3: When line 15 will be executed?

(please tick all that appropriate)

A3: ☐ i = 8; ☐ index = -1;

☐ n[i] = -1; ☐ Never!

☐ Line 14 is TRUE

☐ Line 14 is FALSE

3

Note: Compare it to *"Sorting & Searching: Linear Search"*

Why I was wrong?

____

____

____

____

____

DIVTIC Weekly Task – Week 9: Structures

Name: ______________________________

Student ID: ____________________ **Date:** ____________

Week 9: Structures

/10

Problem 1: From a given code below, please answer the following questions:

```
#include<stdio.h>                                        /* line 1 */
main( ){                                                 /* line 2 */
   struct record {                                       /* line 3 */
                    char  name[10];                      /* line 4 */
                    int   age;                           /* line 5 */
                    float gpa;                           /* line 6 */
   };                                                    /* line 7 */
   struct record  s;                                     /* line 8 */
   printf("Name: ");                                     /* line 9 */
   scanf("%s",s.name);                                   /* line 10 */
   printf("\nAge: ");                                    /* line 11 */
   scanf("%d",&s.age);                                   /* line 12 */
   printf("\nGPA: ");                                    /* line 13 */
   scanf("%f",&s.gpa);                                   /* line 14 */
   printf("\nSize of s: %d bytes", sizeof(s));           /* line 15 */
   printf("\nName: %s  ", s.name);                       /* line 16 */
   printf("Age: %d  GPA: %.2f", s.age, s.gpa);           /* line 17 */
                                                         /* line 18 */
   return 0;                                             /* line 19 */
}                                                        /* line 20 */
```

1

Your Answer

Suppose that user enter ***Kane, 20, 3.25*** for ***name, age,*** and ***gpa*** respectively:

Q1: What is the output of line 15?

A1: ______________________________

Q2: What is the output of line 17?

A2: ______________________________

2

DIVTIC System

Suppose that user enter ***Kane, 20, 3.25*** for ***name, age,*** and ***gpa*** respectively:

Q1: What is the output of line 15?

A1: ______________________________

Q2: What is the output of line 17?

A2: ______________________________

3

Note: Compare it to *"Structures: Creating a Basic Structure"*

Why I was wrong?

____

____

____

____

____

Problem 2: Suppose, we are given the following information:

```
typedef   struct{
                char      n[5];
                int       age;
                char      sex; /* m—male, f—female */
}employee;

employee record[3]={ "Sam",28,'m',   "Ann",25,'f',
                                "Dan",33,'m'};

        employee   *ptr;
```

Write a complete program to count the total of male and female in this record by using a pointer, *ptr*, to access to the array of this record, *record[ ]*, and print it out on the screen as in the following example of a running program.

NOTE: You need to declare more variables where appropriate.

Example of a running program

Male: 2

Female: 1

```
#include <stdio.h>
main( ){

   typedef   struct{
        char     n[5];
        int      age;
        char     sex; /* m-male, f-female */
   }employee;

   employee  record[3]={ "Sam", 28, 'm',
                         "Ann",  25, 'f',
                         "Dan", 33, 'm'};
   employee   *ptr;




   return 0;
}
```

Note: Compare it to
"Structures: Pointer and an array of Structure"

Why I was wrong?

DIVTIC Weekly Task – Week 10: Data Files

Name:____

Student ID:__________________________ **Date:**___________

/10

Week 10: Data Files

Problem 1: Write a complete program to accept two strings from the user (less than 10 characters) and then write them as a text file (Formatted data file) onto a disk in drive **A:**, called **"sample.dat."**

<u>Example of a running program</u>

String1: *Bangkok*
String2: *Korat*

```
#include<stdio.h>
#define NULL 0
main( ){

  char s1[10],  s2[10];
  FILE *fptr;

  fptr = fopen(.............................., "w");

  if(fptr == NULL)
     printf("\nERROR\n");
  else{
        printf("String1: ");
        .....................

        printf("String2: ");
        .....................

        fprintf(.................................);
        fprintf(.................................);
  }
  fclose(......................................);
  return 0;
}
```

Note: Compare it to
"Data Files: Writing a Simple Text File"

Why I was wrong?

Problem 2: Write a complete program to accept data, ***age*** and ***name***, from the user until the user enters ***0*** (zero) for ***age***. Then write a binary file (Unformatted data file) onto a disk in drive **A:**, called "**data.bin**."

Given a structure data type as:

```
typedef struct {
        int  age;
        char n[8];
} record;
```

Example of a running program

Age (enter '0' when finished): *25*
Name: *Kenny*
Age (enter '0' when finished): *19*
Name: *Janet*
Age (enter '0' when finished): *0*

```
#include<stdio.h>
typedef struct {
            int    age;
            char   n[8];
} record;
FILE ..................;
main( ){
  int flag = 1;    record std;
  fptr = fopen(...................................);
  while (flag) {
    printf("\nAge (enter \'0\' when finish): ");
    scanf(.........................................);
    if(std.age == 0)
        break;
    printf("Name: ");
    scanf(.........................................);
    fwrite(.........................................);
    }
    fclose(fptr);
    return 0;
}
```

Note: Compare it to "
"Data Files: Writing a Simple Binary File"

Why I was wrong?

____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____