The Other Art of Computer Programming. Milestone 3: COBOL.
1970s

Melanie Tarr
Edith Cowan University, melanie.tarr@gmail.com

Follow this and additional works at: https://ro.ecu.edu.au/ecuworkspost2013

Part of the Computer Sciences Commons, and the Education Commons

10.25958/5e3a75973d41c
This Book is posted at Research Online.
https://ro.ecu.edu.au/ecuworkspost2013/7442
COBOL

The Other Art of Computer Programming
by Melanie Tarr

1970s

Milestone

COBOL
Learn how to keep records on students within a virtual university

Compiling
Look at a punched card and learn about BCF grammar
Grace Hopper wrote the first compiler and used a computer called the Mark 1. She abstracted software code in English rather than mathematics so other people could program besides mathematicians.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>How a compiler works</td>
<td>4</td>
</tr>
<tr>
<td>Noam Chomsky’s BCF grammar</td>
<td>5</td>
</tr>
<tr>
<td>Structured programming</td>
<td>6</td>
</tr>
<tr>
<td>Introduction to COBOL</td>
<td>7</td>
</tr>
<tr>
<td>Four students example</td>
<td>8</td>
</tr>
<tr>
<td>Rich picture of student information</td>
<td>10</td>
</tr>
<tr>
<td>Student records</td>
<td>11</td>
</tr>
<tr>
<td>Problem scenario 1</td>
<td>16</td>
</tr>
<tr>
<td>The structure of a COBOL program</td>
<td>21</td>
</tr>
<tr>
<td>A COBOL punched card</td>
<td>31</td>
</tr>
<tr>
<td>COBOL coding sheets</td>
<td>33</td>
</tr>
</tbody>
</table>

Note: This lesson may offer useful explanations for the above Digital Technologies curriculum code in the Australian National Curriculum.
the most dangerous phrase in the English language is “we’ve always done it this way.”

Grace knew that a computer was more than a calculator for numbers.

She knew it could also be used to model other symbols like records about people. To do this she needed to make a compiler.

The compiler needed to take the data put into the computer by an operator and change this into information. Although she invented this in the 50s, COBOL was not widely used until the 70s.
HOW A COMPILER WORKS

1. The program is entered into the computer
   - Fetch instruction

2. The program is decoded
   - The program is transformed into machine code

3. The program is transformed into machine code
   - Machine code is temporarily stored

4. Machine code is temporarily stored
   - Calculator

5. The machine code is allocated a place within the main memory to run
   - Brain icon
A notation technique used by a compiler is Backus Naur Form. As the language is context-free - that is - it is not really English and must follow a set of rules to be understood. These rules can be structured. The structure here depicts the BNF form of an address.
Structured Programming used looping and eradicated 'spaghetti' code that is extremely difficult to understand and maintain. The subroutines were contained within a certain section of the program and the lines of code of the subroutine did not jump to other areas. COBOL was a popular language to program payroll systems.
The Mark 1 at Harvard University was the computer Grace used. Von Neumann also used it to solve an equation for the Monte Carlo subroutine (1950s milestone) in World War 2. The tables Charle’s Babbage wanted printed by the analytical engine were finally possible with the Marc 1. The Mark 1 came from the Babbage designs.

The next section starts with a soft systems analysis of a student records system. This context enables us to see how COBOL was and is still used to assist people. COBOL was introduced first as a business language.
Name: Maria Callas  
Major: Music  
Credit Points: 16

Name: Mahatma Ghandi  
Major: Philosophy  
Credit Points: 25

Name: Albert Einstein  
Major: Physics  
Credit Points: 53

Name: Amelia Erhart  
Major: Aviation  
Credit Points: 24

CONSIDER FOUR STUDENTS
That attend a university
There is a lot of information to record about each student.
EACH STUDENT HAS ONE RECORD

administration enters a student’s data

the data is transformed into information

a student’s information record is created

one university has many student records
Each student is represented by a record.

Each record contains fields:
- the student’s name
- completed subjects
- and major

Ghandi
In summary, a university has students, students have records, records have fields.
A COBOL program will produce line by line output on a printer as it processes code.

Once a line is printed the program executes the next statement in the code.
As with the 1950s milestone - the Turing machine, COBOL programming followed the same type of operations. These are input processing and output.
Problem scenario 1

I would like to know all the students who are studying aviation. I would like a report, listing the aviation students.

To produce the report we repeat the following:

- select a record
- check if the student studies aviation
- write the name on the printed report
- put the folder back
TO A COMPUTER SCIENTIST THE LOGIC LOOKS LIKE THIS

1. Start
2. Open files
3. Read first record
4. Write report heading
5. Process records
6. Check if any more data?
7. Yes: Read next record, No: process records
8. Aviation major?
9. True: Write student name, False: Return to process records
10. End
WE CAN CONVERT THIS APPROACH TO A FORM OF VISUAL COMMUNICATION. THE ORDER OF RECORDS THAT THE STAFF MEMBER ENTERS THROUGH THE KEYBOARD IS ALBERT THEN AMELIA.
Most of the logic of the program resides in the procedure division. As the machine operates in sequence it goes through three other divisions before the procedure division. These are the:

- identification,
- environment and data divisions.

These mostly deal with initialisation.
The authors believe, however, that immediate exposure to a real program is extremely beneficial in stripping the mystical aura that too often surrounds programming.

Villar, p. 9

COBOL is intended to resemble English...
MY NAME IS AVIATION PROGRAM
MY CREATOR IS GRACE HOPPER.

the code divisions in a cobol program
Cybernetics is Greek for the art of steering. It is not a quest to make computers smarter. It is as old as nature. It connects control (actions taken in hope of achieving goals) with communication (connection and information flow between the actor and the environment).

Norbert Weiner

Conversations between the program creator (Grace Hopper) and the 1960s thinking machine (Xerox Alto) will be used to explain the code listings like the one below.

```
001 IDENTIFICATION DIVISION.
002 PROGRAM-ID.   AVIATION.
003 AUTHOR.        GRACE HOPPER.
```
WHERE AM I GETTING DATA FROM AND WHERE DO I SEND INFORMATION?

YOU ARE GETTING DATA FROM THE KEYBOARD AND MOUSE. YOU ARE SENDING INFORMATION TO THE PRINTER.

005  ENVIRONMENT DIVISION.
006  INPUT-OUTPUT SECTION.
007  FILE-CONTROL.
008    SELECT STUDENT-FILE
009    ASSIGN TO UT-S-SYSIN
010    SELECT PRINT-FILE
011    ASSIGN TO UT-S-SYSOUT
A CLOSER LOOK AT THE DATA DIVISION (LINES 13-39)

FILE SECTION (LINES 14-21)

THE FILE SECTION TELLS COBOL WHAT LENGTH THE DATA IS COMING IN FROM THE KEYBOARD.

25 characters

15 characters

43 characters

HOW LONG IS THE DATA THAT COMES FROM THE KEYBOARD?

YOUR STUDENT FILE IS 43 CHARACTERS LONG.

FILE SECTION

3 numerals

student name amelia earhart

student credits 126

student major aviation

1970s Milestone three
A CLOSER LOOK AT THE DATA DIVISION
(LINES 13-39)

FILE SECTION
(LINES 14-26)

013 DATA DIVISION.
014 FILE SECTION.
015 FD STUDENT-FILE
016 RECORD CONTAINS 43 CHARACTERS
017 DATA RECORD IS STUDENT-IN.
018 01 STUDENT-IN.
019 05 STU_NAME PIC X(25)
020 05 STU_POINTS PIC 9(3)
021 05 STU_MAJOR PIC X(15).
THE LAST SECTION IN THE FILE SECTION NAMES THE PRINT FILE AND DESCRIBES IT AS 32 CHARACTERS LONG.

023  FD   PRINT-FILE
024  RECORD CONTAINS 132 CHARACTERS
025  DATA RECORD IS PRINT-LINE.
016 01  PRINT-LINE            PIC X(132)
THE WORKING STORAGE SECTION CONTAINS THE LAYOUT FOR THE REPORT HEADINGS

HOW DO I PRINT OUT THE REPORT?

YOU PRINT A HEADING LINE THEN A DETAIL LINE

students that study aviation

heading line

detail line

aviation major 1973
041 PROCEDURE DIVISION.
042 PREPARE-AVIATION-REPORT
043 OPEN INPUT STUDENT-FILE
044 OUTPUT PRINT-FILE

045 READ STUDENT-FILE
046 AT END MOVE ‘NO’ TO IS-THERE-ANymORE-DATA
047 END-READ.

048 PERFORM WRITE-HEADING-LINE.
049 PERFORM PROCESS-RECORDS

055 WRITE-HEADING-LINE.
056 MOVE HEADING-LINE TO PRINT-LINE.
057 WRITE PRINT-LINE.

060 IF STU-MAJOR = “AVIATION”
061 MOVE STU-NAME TO PRINT-NAME
062 MOVE DETAIL-LINE TO PRINT-LINE
063 WRITE PRINT-LINE

065 READ STUDENT-FILE
066 AT END MOVE ‘NO’ TO IS-THERE-ANymORE-DATA.

050 UNTIL IS-THERE-ANymORE-DATA = ‘NO’

051 CLOSE STUDENT-FILE
052 PRINT-FILE.

A CLOSER LOOK AT THE PROCEDURE DIVISION
THE OUTPUT FOR THIS PROGRAM IS AS FOLLOWS.

students that study aviation

aviation major 1973

amelia earhart
COBOL code follows a specific placement on the screen. The structure corresponds to the columns the code is typed into.

<table>
<thead>
<tr>
<th>sequence numbers</th>
<th>comment</th>
<th>area A</th>
<th>area B</th>
<th>program identification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
0015 ENVIRONMENT
0016 CONFIGURATION
0017 SOURCE-COMPUTER
0018 IBM-ES900
0019 OBJECT-COMPUTER
0020 IBM-ES900
0021 INPUT-OUTPUT SECTION
0022 FILE CONTROL
0023 SELECT SALE-FILE ASSIGN
```
COBOL CODING SHEETS
Problem scenario 2

I also want to know the students that have credit values over 100 points as they are eligible for a scholarship.