

AN Introduction to Structured Program Design in COBOL

COBOL : COMMon Business Oriented Language .

Hardware : Any part of a computer system that you can see or touch, such as a keyboard, Screen, Mouse, Joystick, Printer, Speakers... etc.

Software : electronic instructions that tells your computer what to do.
(without the software a computer is like an airplane without a pilot)

Program : Set of instructions that enable a computer to process data.

2 types of computer programs:

1. **Operating system programs**: control the overall operations of the computer. (ex. DOS, Windows).
2. **Application programs**: Perform tasks required by users. (ex. Word, COBOL, Games).

Application Program : Operates on input data and convert it to meaningful output information.

Application program is written by a programmer to provide the users with the information they need.

2 type of applications

1. One time problem solution. (Ex. Program to display an average of set of exams)
2. Application runs periodically or in a regular basis. (Ex. Program to print student's transcripts)

Information systems: A set of computerized business procedures in an application area.

Customized Program: program is written for specific user.

Machine Language Programs:

All programs to be executed by the computer must be in a machine language. Since machine language is so difficult, programming languages were developed to enable the programmer to write English-like or symbolic instructions.

Symbolic Programming Languages:

that can be translated or **compiled** into machine language. Such as COBOL, Pascal, Basic, C or C++.

Compiler : translator program that translates (converts) symbolic languages into machine languages.

- COBOL is extensively for commercial applications.

The Applications Program Development Process:

Programmers who begin with the coding phase often produce poorly designed or inadequate programs.

The following is an overview of the steps involved in program development process.

1. Obtaining Program specifications: getting the system input/output specification for a program.
2. Design Using Program Planning Tools: Design using the following tools
 - Flowcharts
 - Pseudocode
 - Hierarchy charts
3. Code The Program: Write the program source code. (COBOL CODE)
4. Compiling and Testing the program: Make sure that the program has **NO** Syntax or Logic Errors.
5. Documenting the program: write the procedure manual. It is for people who will be working with the output and the operators who will run the program on regularly scheduled basis

The Nature of COBOL:

- **COBOL is Business Oriented Language :**

Used to develop business applications such as payroll, inventory. They typically operate on large volume of data. They not suitable for scientific applications where complex calculations are required.

- **COBOL is a standard Language:**

COBOL program may be compiled and runs on a variety of different machines with COBOL compiler of that machine and minor modifications. Because the language is so widely used, the computers of the next generation will support COBOL.

- **COBOL is English-Like Language:**

Basically, Cobol has 2 advantages:

1. Common to most computers.
2. It is business oriented.

The third advantage is that All instructions can be coded using English words rather than complex codes. (ex. The word Add is used to add 2 numbers.)

- **COBOL is User-Friendly Language:**

Because users are frequently able to understand the English-Like instructions of COBOL, it is considered a user-friendly language. It is not overly technical like other languages such as C/C++.

A History of COBOL and the ANS Versions:

Read from the text book

American National Standard (ANS):

Read from the text book

Techniques for Improving Program Design

1) **Structured Programming Using Modular Design for Coding Paragraph.**

This technique uses logical control constructs that makes the program easier to read, debug, and modify. (Reduce the number of GO TOs in the program) (It is called the GO-TO-Less Programming. Sequence is controlled by the PERFORM Statement. (Same as WHILE .. DO, in PASCAL). Each section of the program is written independently and debugged separately.

A typical structured program is subdivided into paragraphs or **MODULES** where the main module calls in the other modules as needed. Each module can be tested independently.

Modules can be written by different people and the main module calls those modules.

2) **A Top-Down Approach for Coding Modules.**

Develop the major modules before going the minor ones. Example is the same as writing a term paper. One starts with the outline then takes each point in the outline and expands it.

A Sample Program

An Overview of the four Divisions of the any COBOL Program

Name	Purpose
IDENTIFICATION DIVISION	Identifies the program to the computer. It also provides some documentation about the program
ENVIRONMENT DIVISION	Describes the specific computer equipment that will be used by the program
DATA DIVISION	Describes the input and the output formats to be used by the program. It also defines any constants and work areas necessary for the processing of data.
PROCEDURE DIVISION	Contains the instructions necessary for reading input, processing it, and creating the output

The IDENTIFICATION and ENVIRONMENT DIVISIONS.

The Identification Division has ONE required entry. The required entry is Program-ID. This to serve as a name of the program. There are other optional entries such as Author, Date Written, and Date Compiled.

Syntax is

Identification Division.

Program –ID. Sample.

Author. Husain Ghloom

Date-Written. 03/07/2009.

Date-Compiled. 03/07/2009.

The Environment Division.

The Environment Division assigns the input and the output **files** to specific devices in the e INPUT-OUTPUT Section.

Syntax is

Environment Division.

Input-Output Section.

File-Control.

Select *InputFilename* **Assign To Disk.**

Select *OutputFileName* **Assign To Sysoutput.**

The Data Division.

The FILE Section of the Data Division describes the format of the input and the output files. The data division has e Sections: The File-Section and the Working-Storage Section.

1) **File Section:** The **FD** (*File Description*) Describes the Input and the output files. Each file has a label. Within each file, there is a record description. Each record contains filed names , their types, and their sizes.

2) The **Working-Storage Section.** This section is used to define any field that is not part of the input or the output record.

Syntax is

Data Division.

File Section.

FD *InputFileName.* **Label Records are standard.**

```
01 input Record.
   05 Filed1                PICTURE X(5).
   05 Filed2                PICTURE 9(5).
   05 Field3                PICTURE 99v999.
```

.....

...

...

FD *OutputFileName.* **Label Records are Omitted.**

```
01 OutputRecord.
   05 OutputFiled1 PICTURE X(5).
   05 OutputFiled2 PICTURE 9(5).
   05 OutputFiled3 PICTURE 99.999.
```

.....

...

...

Working-Storage Section.

```
01 ARE-THERE -MORE-RECORDS    PIC XXX Value 'YES'.
```

The Procedure Division.

The Procedure Division contains the set of instructions to be executed by the computer. Each instruction is executed in the order in which it appears in the coding sheet. First, a file must be opened, Then records are read, and operations are performed on each record. When all records are completed, the input and the output files must be closed and the program is terminated by a stop run command.

Syntax is

```

Procedure Division
Open Input   InputFileName
Output   OutputFileName.
Read   InputFileName   At End XXXXXXXXXXXXXXXX
Perform Paragraph-A.
Close   InputFileName
          OutputFileName.
Stop Run.
    
```

.....

.....

Example

A computer center of a large company is assigned the task of calculating weekly wages or gross pay for all nonsalaried personal. The employee name, hourly rate, and number of hours worked are supplied as an input for each employee, and the weekly wages figure is to be computed as follows:

$$\text{WEEKLY-WAGES} = \text{HOURS-WORKED} \times \text{HOULY -RATE.}$$

Input Layout

Employee Name	Hours Worked in A Week	Hourly Rate	
		Dollars	Cents
1	20 21	22 23	24 25

Output Layout

```

XXXXXXXXXXXXX           99           9.99           999.99
(Name Out )           Hours-Out       Rate Out       Weekly Wage
    
```