UNIT 1

Introduction To Language Features

Common Business Oriented Language

- 1959 – New Language is named COBOL
- 1960 – Codasyl established COBOL maintenance committee
- 1961 – 1st version of complier made available. Users started writing programs
- 1968 –2nd version of cobol was approved and standardized by ANSI
- 1974 – Revised and released as COBOL-74
- 1985- Revised and released as COBOL-85

Figure 1-1 Common Business Oriented Language

To meet the increasing demands for a high level language suitable for business data processing, the United States Department of Defense Convened a Conference on 28th and 29th of May 1958.
Three committee were formed for the actual design of the language.
In September 1959 the short term committee submitted a report to the Defense Directorate thus COBOL came into existence.
COBOL is known as a structured programming language because it allows programmers to segregate the modules and put them into different paragraphs in a more efficient way.
Some of the features of COBOL are
It is English-like and more easily readable
Efficient file handling capabilities.
More than 70% of business applications are running on COBOL
Reduces the efforts required for documentation of the program.
The following features are available with VS COBOL II:

MVS/XA and MVS/ESA support
The compiler and the object programs it produces can be run in either 24- or 31-bit addressing mode

COBOL Program Organization

IDENTIFICATION DIVISION
ENVIRONMENT DIVISION
DATA DIVISION
PROCEDURE DIVISION

Figure 1-2 COBOL Program Organization

Notes:

The four divisions of the COBOL source program are :

- **IDENTIFICATION DIVISION**

  This division's primary purpose is to name the program

- **ENVIRONMENT DIVISION**

This division is primarily used to tell the computer about the input and output devices such as files or printers.

- **DATA DIVISION**

  This division describes the data the program will be using and carves out sections of memory to map the data. Here you would distinguish between data, which will be used for a " scratch pad" area called WORKING-STORAGE and the holding area for data that will be used by the files.

- **PROCEDURE DIVISION**

  The PROCEDURE DIVISION is the section of our program where the logic or commands reside. This is the logic or rules we will use to manipulate the data defined in the DATA DIVISION to solve a business problem.

Cobol Language Structure

- **Characters**
- **Character String**
- **COBOL Words**
- **User-Defined Words**
- **Reserved Words**
- **Figurative Words**
- **Special Registers**
- **IBM Extensions**
- **Non-numeric and numeric Literals**

Figure 1-3 COBOL Language Structure

Notes:

Structure of a Cobol Program

	Examples
• **Divisions**	DATA DIVISION
• **Sections or Paragraphs**	PROGRAM-ID FILE SECTION, 100-PARA
• **Statements**	MOVE A TO B
• **Sentences**	IF A>B MOVE A TO B ELSE ADD C TO D

Figure 1-4 Structure of a Cobol Program

Notes:

All COBOL programs should follow the structure. Rules of coding varies, depending on the compiler versions but the structure remains same. A period (.) is a must at the end of each sentence and indicates the end of the sentence.

Character Set of COBOL

COBOL supports the following characters

Numbers	:	0-9
Alphabets	:	a-z, A-Z
Spaces or blanks		
Arithmetic operators	:	ex: **, *, +, -, /
Special characters	:	ex: - \ / , ;

Figure 1-5 Character Set of COBOL

Notes:

The COBOL dictionary words used for coding are called COBOL reserved words and they should not be used as user-defined words.
Lower case alphabets can be used for coding depending on the compiler version.
Comma (,) or space is used as separators for user-defined words.

Sample COBOL Program

```
Columns
1       6 7  8   11 12                                          72 73        80

          *  This is a sample program

          IDENTIFICATION DIVISION.
          PROGRAM-ID. SAMPLE.
          ENVIRONMENT DIVISION.
          DATA DIVISION.
          WORKING-STORAGE SECTION.
          01    A    PIC  9(2)  VALUE  20.
          01    B    PIC  9(2)  VALUE  3O.
          01    C    PIC  9(3)  VALUE  ZEROS.
          PROCEDURE  DIVISION.
                    DISPLAY  'THE SUM IS'.
                    ADD   A ,B  GIVING C.
                    DISPLAY C.
                    STOP RUN.
```

Figure 1-6 Sample COBOL program

Notes:

1-6 ---------- Sequence numbers
7 ---------- Indicator/Comment/Continuation
8-11 ---------- Area A
12-72 ---------- Area B
73-80 ---------- Descriptor

This foil shows a sample COBOL program to 'ADD' two numbers and 'DISPLAY' the sum. SAMPLE is the program name.
SAMPLE, A, B AND C are called user-defined words.
A, B,C are called variables or data-items.

Coding Format

Columns	Name	Purpose
01-06	Sequence	Sequence numbers are generated by Cobol Compiler for each line.
07	Indicator	To mark an asterisk (*) or a slash (/) for comment line, or a hyphen (-) for continuation of a statement.
08-11	Area A	All division headings, section and paragraph headings and '01' level entries should begin from this area.
12-72	Area B	All Cobol statements and sentences should lie within this area
73-80	Description	Any thing written in this area is ignored.

Figure: 1-7 Coding Format

Notes:

COBOL coding should follow the standard format.
The Screen is divided into different areas for the purposes explained above.
All statements indicating action are called COBOL verbs and should begin from 12th column or after.
-E.g MOVE, ADD, DIVIDE, STOP RUN

User-defined Words

Valid	Invalid	Reason
TOTAL-OF-FIGURES	DATA	Cobol reserved word
34B100-PARA1	-48B	Hyphen in beginning
GROSS-PAY	GROSS PAY	space in b/w 2 words

Literals	Examples
Numeric constants	35, -345.67
Alphanumeric constants	'Leo talstoy'
	'ka01-h215'

- Paragraph names, Identifiers, File names can be defined by users.
- The terms identifiers, data-names, variables, data-items are often used interchangeably indicates memory.

Figure: 1-8 User Defined Words

Notes:

All user-defined words should conform to following rules

- Length should not exceed 30 characters.
- At least one character must be an alphabet.
- Spaces and special characters are not allowed.
- Word can contain hyphens (-) but not in the beginning or at the end
- Cannot be a COBOL reserved word.