COBOL: Common Buisness Oriented Language
Divisions - IDENTIFICATION, ENVIRONMENT, DATA & PROCEDURE
Working:

- computer only understand binary language
- COBAL code is converted into binary language using a Compiler
- Compiler checks the syntax and convert into Binary language
- Compiles creates an output file which is known as Load module

Importance:

1. first widely used high level programming language
2. user friendly, can be coded in simple english language
3. used as self - documenting language
4. can handle huge date processing
5. highly Compatible and easier debugging

Features:

1. Standard Language - can be compiled and run on machines like IBM AS /400, personal computers
2. Business Oriented - designed for business oriented applications related to financial domain, defense domain etc.
3. Robust Language - has numerous testing and debugging tools available for almost all computer platforms
4. Structured Language - has logic control structures, makes it easier to read and modify, different divisions so it's easy to debug

Structures:
A COBOL program structure consists of divisions as shown

Program - > Divisions - > Sections - > Paragraphs - > Sentences - > Statements - > Characters

Sections - collection of paragraphs, logical subdivision of program logic
Paragraphs - Subdivision of a section or division

- either user defined or pre-defined name followed by period
- Consists of zero or more sentences / entries

Sentences - combination of one or more statements

- appear only in procedure division
- must end with a period

Statements - meaningful COBOL statements that perform some processing .
Characters - the lowest in the heirarchy and cannot be divisible
example code:

```
PROCEDURE DİVISION,
A0000-FIRST-PARA SECTION,
FIRST -PARAGRAPH,
ACCEPT WS-ID
MOVE ' 10 ' TO WS-ID
DISPLAY WS-ID
.
```

} Sentence

Statement

Divisions:
COBOL has four divisions

1. Identification divisions - only mandatory division, used to identify the program. PROGRAM -ID - only mandatory paragraph, specifies the program name that can consists 1 to 30 characters.

2. Environment Divisions - used to specify input and output files to the program.

   Consists of two sections :-

   Configuration Section - provides information about the system on which the program is written and executed.

   It consists of two paragraphs :-

   Source computer - System used to compile the program

   Object computer - System used to execute the program

   Input-Output section - provides information about files to be used.

   It consists of two paragraphs :-

   File Control - Provides information of external data sets used in the program

   I-O Control - provides information of file used in the program

3. Data division - used to define the variables

   consists of four sections :-

   File Section - defines record structure of the file

   Working - Storage section - declare temporary variables and file structures

Local-Storage section - allocates variables every time a program starts
Linkage Section - to describe the data names that are received from an external program

4 . Procedure Division - used to include the logic of the program

- Consists of executable statements using Variable defined
- paragraph and sections names are user - defined
- must be atleast one statement
    - STOP RUN - last statement to end the execution in calling programs
    - EXIT PROGRAM - last statement to end the execution in called programs

Character Strings: Comment, Literal, COBOL word.
Data division: Data Name, Level number, Picture clause, Value clause
Picture Clause used to define Data type, sign, Decimal point position and Length
COBOL verbs: used in the procedure division for data processing.
Input/Output verbs - ACCEPT, DISPLAY, INITIALIZE, MOVE

Data Layout: COBOL Layout is the description of use of each field and the values present in it.
REDEFINE, RENAMES, USAGE clauses
USAGE CLAUSE:
IS DISPLAY - data is stored in ASCI format

IS COMP -

COMP-1: Real or Float

COMP-2: Long or Double

COMP-3: Decimal

Copybooks: a selection of code that defines data structure

Conditional Statements:

IF Conditional statement - END-IF, Nested-IF

Relation conditional statement -

Equal to (=), Greater than or Equal (> =), Lesser then or Equal (<=), Greater than (>), Less than (<)

Sign Condition - IS, NOT, Positive, Negative, Zero.

Class condition - NUMERIC, ALPHABETIC, ALPHABETIC - UPPER, ALPHABETIC-LOWER.

Condition-name condition - a user defined name. It behaves like Boolean variables.

syntax:

88 [condition name] VALUE [IS, ARE] [Literal] [THRU LITERAL]

Negated condition: given by using NOT keyword. If the condition is true and NOT in front of it ,then it's final value will be false.

Combined condition: Contains two or more conditions using logical operators AND or OR

Evaluvate Verb: used to evaluate more then one condition. Similar to SWITCH statement in C.

Loop Statements:

Perform thru - used to execute a series of paragraph by giving the first and last paragraph names in the sequence

In - line Perform: Statements inside the PERFORM will be executed till the END-PERFORM is reached
Out - of - line PERFORM: A statement is executed in one paragraph and then the control is transfered to other paragraph or section

Perform Untill - a paragraph gets executed until the given condition becomes true.
Perform Times - a paragraph will be executed the number of times specified
Perform varying - a paragraph will be excited till the condition in untill phrase becomes true

GO TO Statement:

- used to change the flow of execution in a program.
- transfer goes only in the forward direction
- used to exit a paragraph

Unconditional Go To - Syntax: Go To para-name
Conditional Go to - Syntax: Go To para-1 para- 2 para-3 DEPENDING ON x

String Handling:

- used to do multiple functional operations on strings.

Handling Statements -

Inspect - used to count or replace the characters in a string

Tallying - used to count the string characters
Replacing - used to replace the String characters

String - used to concatenate the strings
Unstring - used to split one string into multiple sub - strings

Table Processing:

- Arrays in COBOL are called as Tables.
- An array is a linear data structure and is a collection of individual data items of same type.
- Data items of Tables are internally stored

  Table Declaration:

  - Table is declared in Data Division
  - Occurs clause is used to define a table

    One - Dimensional Table - occurs clause is used only in declaration

    Two - Dimensional Table - created with both data elements being variable length

  Subscript:

  - Table individual elements can be accessed by using subscript
  - value ranges from 1 to the number of times the table occurs
  - it can be a positive number
  - doesn't require a declaration in data division
  - automatically created with occurs clause

  Index:

  - Table elements can also be accessed using index
  - It is a displacement of an element from the start of the table
  - declared with Occurs clause using INDEXED BY clause
  - The value of index can be changed using SET statement and PERFORM Varying options

  Set Statement:

  - used to change the index value

- Set verb is usual to initialize, increment, or decrement the index value
- used with Search and search All to locate elements in table

Search:

- a linear Search method used to find elements inside the table
- performed on both sorted and unsorted tables

Search All:

- a binary search method used to find elements inside the table
- doesn't require initialization

File Handling:

- PS (Physical Sequential) and VSAM files are used in COBAL

Field - used to indicate the date stored about an element
Fields can have following attributes:

- Primary keys - folds that are unique to each record and are used to find a particular record
- Secondary keys - unique or non unique fildes that are used to search for related data
- Descriptors - fields are used to describe an entity

Record - a collection of fields that is used to describe an entity
The cumulative size of all the fields in a record is known as the record size
Physical record - the information that exists on the external device. It is also known as a block.

Logical record - the information used by the program.

- In COBOL programs, only one record can be handled at any point of time and it is called as logical record

File - a collection of related records.

File Organization:

- Indicates how records are organized in a file

Types of File Organization Schemes:

Sequential File Organization - A Sequential file consists of records that are stored and accessed in sequential order. key attributes:

- records can be read in sequential order
- records are written in sequential order. A new record cannot be inserted in between and is always inserted at the end of the file
- After placing a record in a Sequential file, it is not possible to delete, shorten, or lengthen a record
- Order of the records, once inserted, can never be changed
- Updation of record is possible.A record can be Overwritten, if the new record length is Same as the old record length
- Sequential output files are good option for printing

Indexed Sequential File Organization - An indexed sequential file consists of records that can be accessed sequentially. Direct access is also possible. It consists of two parts:

- **Data File** contains records in sequential scheme.
- **Index File** contains the primary key and its

address in the data file

key attributes -

- records can be read in sequential order
- records can be accessed randomly if the primary key is known. Index file is used to get the address of a record and then the record is fetched from the data file
- Sorted index is maintained in this file system which relates the key value to the position of the record in the file
- Alternate index can also be crated to fetch the records

Relative File Organization - A relative file consists of records ordered by their relative address.

key attributes:

- records can be in sequential order
- records can be accessed using Relative key. **Relative key** represents the records location relative to the address of the start of the file
- records can be inserted using relative key. Relative address is calculated using relative key
- Relative file provides the fastest access to the records
- The main disadvantage of this file system is that if some intermediate records are missing, they will occupy space.