UNIT 2

The Organization of a COBOL Program

IDENTIFICATION DIVISION

IDENTIFICATION DIVISION.	Required
PROGRAM-ID. <Pgm-name>	Required
AUTHOR. <Pgmr-name>	Optional
DATE –WRITTEN. <Entry>	Optional
DATE-COMPILED. <Entry>	Optional
SECURITY. <Entry>	Optional

At least one space required after the period

Figure 2-1 IDENTIFICATION DIVISION

Notes:
The Identification Division must be the first division in every COBOL source program. It must be coded as IDENTIFICATION DIVISION or ID DIVISION followed by a separator period.

ENVIRONMENT DIVISION

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. <Entry>.
OBJECT-COMPUTER. <Entry>.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
         ---------------------------
         ---------------------------
I-O-CONTROL.
         ----------------------------
         ----------------------------
```

Figure 2-2 ENVIRONMENT DIVISION

Notes:

The Environment Division is divided into two sections:

- **The CONFIGURATION SECTION**

The Configuration Section is an optional section for programs which describe the computer environment on which the program is compiled and executed.
The Configuration Section can be specified only in the ENVIRONMENT DIVISION of the outermost program of a COBOL source program.

- **The INPUT-OUTPUT SECTION**

The Input-Output Section of the Environment Division contains two paragraphs:

  - FILE-CONTROL paragraph
  - I-O-CONTROL paragraph
  - **FILE-CONTROL paragraph**

The keyword FILE-CONTROL can appear only once, at the beginning of the FILE-CONTROL paragraph. It must begin in Area A, and be followed by a separator period. The FILE-CONTROL paragraph is optional.

The FILE-CONTROL paragraph associates each file in the COBOL program with an external data set, and specifies file organization, access mode, and other information.

There are three formats for the FILE-CONTROL paragraph:

- QSAM, SAM, and VSAM sequential file entries
- VSAM indexed file entries
- VSAM relative file entries.

The FILE-CONTROL paragraph begins with the word "FILE-CONTROL", followed by a separator period. It must contain one and only one entry for each file described in an FD or SD entry in the Data Division. Within each entry, the SELECT clause must appear first, followed by the ASSIGN clause. The other clauses can appear in any order.

- **I-O-CONTROL paragraph**

Specifies information needed for efficient transmission of data between the external data set and the COBOL program. The series of entries must end with a separator period

The keyword I-O-CONTROL can appear only once, at the beginning of the paragraph. The word I-O-CONTROL must begin in Area A, and must be followed by a separator period.

Each clause within the paragraph can be separated from the next by a separator comma or a separator semicolon. The order in which I-O-CONTROL paragraph clauses are written is not significant

DATA DIVISION

Data division is the third and most frequently used division in all programs. Every variable required by the program should be declared in appropriate section of the data division, before using in procedure division

The Data Division is divided into three sections:

- File Section
  Defines the structure of data files (including sort-merge files).

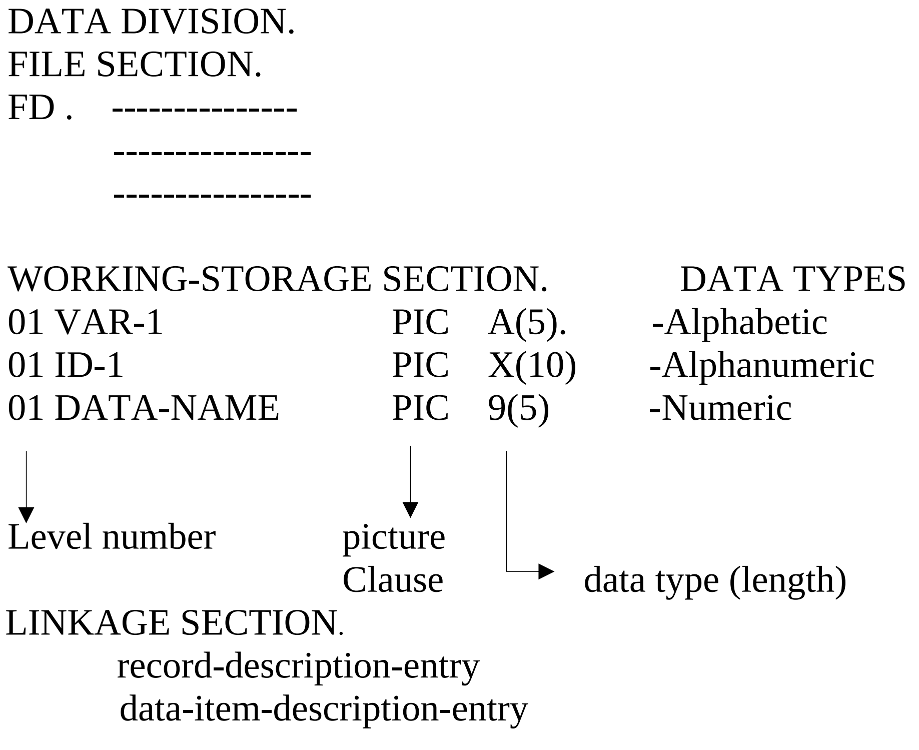
- Working-Storage Section

  Describes records and subordinate data items that are not part of data files but are required by the program.

- Linkage Section

  Describes data made available by another program. It usually appears in the called program and describes data items that are referred to by the calling and the called programs.

Each section has a specific logical function within a COBOL source program, and each can be omitted from the source program when that logical function is not needed. If included, the sections must be written in the order shown.



```
DATA DIVISION.
FILE SECTION.
FD .    ---------------
        ----------------
        ----------------

WORKING-STORAGE SECTION.              DATA TYPES
01 VAR-1              PIC   A(5).     -Alphabetic
01 ID-1               PIC   X(10)     -Alphanumeric
01 DATA-NAME          PIC   9(5)      -Numeric

Level number      picture
                  Clause                data type (length)
LINKAGE SECTION.
        record-description-entry
        data-item-description-entry
```

Figure 2-3 DATA DIVISION

Notes:

DATA-ITEMS

- **Explicitly identifies the data being described**
- **The data-item must be the first word following the level-number.**
- **The data-item values can be changed during program execution.**
- **A data-item name cannot be the same as a section-name or a paragraph name**

Figure 2-4 DATA-ITEMS

Notes:

Data item is a user-defined word which is associated with Level number.

COBOL Reserved words should not be Data items.

Level Numbers

- **Range of level numbers available are 01 to 49**
  and
  66 level specified for RENAMING CLAUSE

  77 levels specified exclusively for elementary items

  88 levels specified for CONDITION NAMES.

- **An elementary item can be declared with level numbers 01 and 77**
- **01 and 77 level entries must begin from area A and other level entries can begin from any where in area A or area B**

Figure 2-5   Level Numbers

Notes:

Level represents the nature of a data item.
The level-number specifies the hierarchy of data within a record, and identifies special-purpose data entries. A level-number begins a data description entry, a renamed or redefined item, or a condition-name entry. A level-number has a value taken from the set of integers between 01 and 49, or from one of the special level-numbers, 66, 77, or 88.

Level-number 01 and 77 must begin in Area A and must be followed either by a separator period; or by a space, followed by its associated data-name, FILLER, or appropriate data description clause.

Level numbers 02 through 49 can begin in Areas A or B and must be followed by a space or a separator period.
Level number 66 and 88 can begin in Areas A or B and must be followed by a space.

Single-digit level-numbers 1 through 9 can be substituted for level-numbers 01 through 09.

Successive data description entries can start in the same column as the first or they can be indented according to the level-number. Indentation does not affect the magnitude of a level-number.

When level-numbers are indented, each new level-number can begin any number of spaces to the right of Area A. The extend of indentation to the right is limited only by the width of Area B.

Higher numbered level(s) represent subordinate definition(s).

Level numbers need not be consecutive(but should be in ascending order)

Special Level Numbers

LEVEL-66 contains a RENAMES clause. It regroups previously defined names

LEVEL-77 defines ELEMENTARY items with no subdivision and are unique

LEVEL-88 establishes condition-name entries, associated with a VALUE clause

```
66  data-name-1  renames-clause.

88  condition-name-1 value-clause.
```

Figure 2-6 Special Level Numbers

Notes:

LEVEL-66 regroups previously defined items.

A level-66 entry cannot rename another level-66 entry, nor can it rename a level-01, level-77, or level-88 entry.

All level-66 entries associated with one record must immediately follow the last data description entry in that record.

LEVEL-77 items are ELEMENATARY items with no subdivision. LEVEL-77 names are unique because they can not be qualified.

LEVEL-88 describes condition-names.
LEVEL-88 can be used to describe both elementary and group items.

Picture Clause

Describes the characteristics of the data

CODE	**meaning**
A	alphabetic or space
B	Blanks or spaces
G or N	Graphical data
9	Indicates a Numeric
X	Indicates an Alpha Numeric
P	Indicates the position of the assumed decimal point when the point lies outside the data item.
V	Indicates the position of assumed decimal point of numeric field.
S	Indicates whether the data item signed.

Figure 2-9 Picture Clause

Notes:
Picture clause specifies the data type of an identifier.

Identifier with PIC clause ‘9’ implies that it is **numeric data type**, which can take art in arithmetic computations. ‘V’ and ‘S’ clauses are allowed with numeric data types only.

‘X’ clause represents an **alphanumeric data type** which can hold any character including numbers also.

‘A’ clause indicates an **alphabetic data type**.
Group items are always considered as alphanumeric only. Therefore GROSS-PAY, DEDUCTIONS can not be used for computations.

W-S Declarations

```
WOKING-STORAGE SECTION.

01 PAY.
       05 GROSS-PAY.                                  Alternatively
              10 BASIC    PIC    9(4)V99.
              10 DA       PIC    9(4)V99.            9(4)V9(2)
              10 HRA      PIC    9(4)V99             9999V99

       05 DEDUCTIONS.
              07 PF-DED   PIC    9(3)V99.
              07 IT-DED   PIC    9(3)V99.
       05 NET-PAY         PIC    9(4)V99.
       05 NAME            PIC    A(5).               AAAAA
       05 E-CODE          PIC    X(6).               XXXXXX
```

Figure 2-7 W-S Declarations

Notes:

Pay, gross-pay, deductions are called group items and they don’t have PICTURE clause. Other elements with picture clause are called elementary items, which cannot be broken further.

Pay is a Group item is divided into Gross-pay, Deductions, net-pay, name, e-code further Gross-pay sub-divided into Basic, DA, HRA and DEDUCTIONS sub-divided into PF-DED and IT-DED.

FILLER

FILLER is a COBOL Reserved Word used to describe data fields that will not be referenced in the PROCEDURE DIVISION.

If the data-name of FILLER clause is omitted, the data item being described is treated as though it was FILLER

```
01 EMPLOYEE-RECORD.
   05  EMPLOYEE-TYPE            PIC X.
   05  EMPLOYEE-SERIAL          PIC X(6).
   05  EMPLOYEE-NAME            PIC X(30).
   05                           PIC X(2).
   05  EMPLOYEE-ADDRESS         PIC X(60).
   05  FILLER                   PIC X(34).
```

Figure 2-8   FILLER

Notes:

FILLER is a data item that is not explicitly referred to in a program. The key word FILLER is optional. If specified, FILLER must be the first word following the level-number.

IF data-name or FILLER clause is omitted, the data item being described is treated as though FILLER had been specified.

The VALUE clause may be used on FILLER items, e.g. to assure BLANKS in header lines between fields.

In a MOVE CORRESPONDING statement ,or in an ADD CORRESPONDING or SUBTRACT CORRESPONDING statement ,FILLER items are ignored.

In an INITIALIZE statement, elementary FILLER items are ignored.

USAGE Clause

```
<level number> data-name    [PIC        X(n)]    [USAGE]   COMP
                                                         COMP-1
                                                              COMP-2
                                                         COMP-3
```

COMP	**-**	**Binary Representation**	**Size:**	**Half/Full/Double word**
COMP-1	**-**	**Hexa Decimal Representation** **for Float**	**Size:**	**Full word**
COMP-2	**-**	**Hexa Decimal Representation** **for Float**	**Size:**	**Double word**
COMP-3	**-**	**Packed Decimal Representation** **Where n is number of digits.**	**Size:**	**round(n/2)+1**

Figure 2-10 USAGE Clause

Notes:

The USAGE clause can be specified for a data description entry with a level-number other than 66 or 88. However, if it is specified at the group level, it applies to each

Elementary item in the group. The usage of an elementary item must not contradict the usage of a group to which the elementary item belongs.

The USAGE clause specifies the format in which data is represented in storage. The format can be restricted if certain Procedure Division statements are used.

When the USAGE clause is not specified at either the group or elementary level, it assumed that the usage is DISPLAY.

Computational (COMP) Usage

When usage is specified as COMP, the numeric data item is represented in pure binary. The item must be an integer(no assumed decimal point is allowed). Such that data items are often used as subscripts.The PICTURE of a COMP item should not contain any character other than 9, S.

COMPUTATIONAL-1 (COMP-1) Usage

If the usage of a numeric data item is specified as COMP-1, it will be represented in one word in the floating –point form. The number is actually represented in Hexa decimal (base 16). Such representation is suitable for arithmetic operations. The PICTURE clause cannot be specified for COMP-1 items.

COMPUTATIONAL-2(COMP-2)Usage

This usage is same as COMP-1, except that the data is represented internally in two words. The advantages is that this increases the precision of the data which means that more significant digits can be available for the item. The PICTURE clause cannot be specified for COMP-2 items.

COMPUTATIONAL-3(COMP-3)Usage

In this form of internal representation the numeric data is the decimal form, but one digit takes half-a-byte. The sign is stored separately as the right most half –a-byte regardless of whether S is specified in the PICTURE or not. The hexa decimal number C or F denotes a positive sign and the Hexa –decimal number D denotes a negative sign. Inorder that data fields can start and end on byte boundaries, numbers with an even number of digits are stored with an extra half-byte of zeroes on the left hand side. Thus an item with

PICTURE S9(5)V9(3) USAGE IS COMP-3

will require 5 bytes to be stored internally. Only the characters 9,S, V and P can be used in the PICTURE of a COMP-3 item.

Value Clause

- **Value Clause defines the initial value of a data item**
- **Must not be used for items declared in FILE SECTION.**
- **Can also specify FIGURATIVE CONSTANTS.**
- **If defined at the group level can be used for array declaration also**

EXAMPLES.

```
01 NUM-1      PIC 9(3)     VALUE 245.
01 E-CODE     PIC X(6)     VALUE “E10K3”.
```

```
At group level                                                  contents

        01 GROUP-ITEM        VALUE IS ‘ER34155’
                05   E-ITEM-1          PIC  X(2).               ‘ER’
                05   E-ITEM-2          PIC  XXX                 ‘341’
                05  E-ITEM-3           PIC  X(3)                ‘55’
```

- **Group item is considered as alphanumeric.**

Figure 2-11 VALUE Clause

Notes:

Assigning values to identifiers is called initialization. If variables are not initialized, then they may contain any value, which was stored at the time of last execution of program. It is advised to always initialize working-storage variables.

REDEFINES Clause

Two or more data items can share the same working storage area by **REDEFINING** a storage area.

Level number data name-1 REDEFINES data-name-2

- Level numbers of data-name-1 and data-name-2 must be identical
- The redefines clause must immediately follow data-name-I
- must not be used for level number 66 or 88 items.
- Data-name-1 should not contain VALUE clause
- Multiple redefinition is allowed

Figure 2-12 REDEFINES Clause

Notes:

- Two or more storage areas defined in the data sometimes may not be used simultaneously, in such cases, only one storage area can serve the purpose of two or more areas if the area is defined.
- The REDEFINES clause used allows the said area to be referred to by more than one data name with different sizes and pictures.

ILLUSTRATES REDEFINES CLAUSE

```
DATA DIVISION.

WORKING-STORAGE SECTION.

01      X1
        02      Y           PIC    99.
        02      Y1          REDEFINES  Y          PIC    XX.
01      X3
        02      Z           PIC X        VALUE “M”.
        02      ZZ          PIC X (25)   VALUE ALL “*”.
        02      ZZZ             PIC X (45)   VALUE ALL "- ".
01      X4      REDEFINES X3.
        02      FILL1           PIC X.
        02      FILL2           PIC X (70).
01      X5      REDEFINES X4.
        02      BUFFER          PIC X (71).


PROCEDURE DIVISION
PARA 1.

        MOVE 20 TO Y.
        DISPLAY X1.
        MOVE "A1" TO Y1.
        DISPLAY X1
        DISPLAY X3.
        DISPLAY X4.
        DISPLAY X5.
        STOP RUN.
```

Duplicate Data Names

- Are allowed, provided they belong to a group item

```
01 Pay-Rec.
        02 Id-numbers                   PIC 9(5).
        02 Name                         PIC X (25).
        02 Dept                         PIC X (20).
01 Print-Rec.
        02 Filler                       PIC X (5).
        02 Id-numbers                   PIC X (5)
        02  Filler                      PIC X (5).
        02  Name                        PIC X (25).
        02  Dept                        PIC X (920).
```

MOVE Id-Numbers (OF | IN) Pay-Rec TO Id-Numbers (OF | IN)Print-Rec.

Figure 2-14 DUPLICATE data names

*** OF and IN are called Qualifiers.**

To move the data stored in the four fields of Pay-Rec. the four MOVE statements serve the purpose.

Using the MOVE CORRESPONDING statement the same can be accomplished.

RENAMES Clause

Syntax:

66 data-name-1 RENAMES data-name-2 THRU data-name-3

E.g. :

```
01      PAY – REC.
        02 FIXED-PAY.
                05 BASIC                PIC    9(6) V99.
                05 DA                   PIC    9(6) V99.
        02 ADDITIONAL-PAY.
                05 HRD                  PIC    9(4) V99.
                05 INCENT               PIC    9(3) V99.
        02 DEDUCTIONS.
                05 PF                   PIC    9(3) V99.
                05 IT                   PIC    9(4) V99.
                05 OTHER                PIC    9(3) V99.
66      PAY-OTHER-THAN-BASIC RENAMES DA THRU INCENT.
66      IT-AND-PF-DEDUCTIONS RENAMES PF THRU IT.
```

Figure   2-15   RENAMES Clause

Notes:
In order to re-group elementary data items in a record, so that they can belong to the original as well as to the new group, the RENAMES clause is used.

LEVEL-66 regroups previously defined items.

A level-66 entry cannot rename another level-66 entry, nor can it rename a level-01, level-77, or level-88 entry.

All level-66 entries associated with one record must immediately follow the last data description entry in that record.

ILLUSTRATES RENAMES CLAUSE

```
DATE DIVISION.
WORKING-STORAGE SECTION.
01      PAY
02      FIXED-PAY
        10      E-BASIC                         PIC 9(6). 99
        10      E-DA                            PIC 9(6). 99.
05      ADDL-PAY.
        10 HRA                                  PIC 9(4). 99.
        10 INCENTIVE                            PIC 9(3). 99.
05      DEDUCTIONS.
        10      E-PF                            PIC 9(3). 99.
        10      E-IT                            PIC 9(4). 99.
        10      OTHERS                          PIC 9(3). 99.
66      PAY-LESS-BASIC RENAMES E-DA THRU INCENTIVE.
66      IT-AND-PF RENAMES E-PF THRU E-IT.

PROCEDURE DIVISION.

MAIN-PARA
        MOVE-123456.78 TO E-BASIC.
        MOVE 234567.89 TO E-DA.
        MOVE 1234.56 TO HRA.
        MOVE 123.45 TO INCENTIVE.
        MOVE 123.45 TO E-PF.
        MOVE 1234.56 TO E-IT.
        MOVE 123.45 TO OTHERS.
        DISPLAY PAY.
        DISPLAY FIXED-PAY.
DISPLAY ADDL-PAY.
DISPLAY DEDUCTIONS.
DISPLAY PAY-LESS-BASIC.
DISPLAY IT-AND-PF.
STOP RUN.
```

Figurative Constants

- Constants frequently used by most programs

Figurative Constants	**Meaning**
HIGH-VALUE(S)	Represents the highest and lowest
LOW-VALUES (S)	value in the collating sequence.
ZERO, ZEROS, ZEROES	One or more Zeroes
SPACE (S)	One or more blanks

Example 01 ID-1 PIC X(3) VALUE SPACES.

- Collating sequence is the order in which the characters are compared by the system.

Figure 2-16 Figurative Constants

Notes:

Figurative constants are reserved words that name and refer to specific constant values.

Edited Fields

- **Move 345.46 to a field of picture 9(3)v99 & display or print You may see different number in result**
- **Characters must be edited before report is taken to suppress leading zeros, to include currency signs or to include date separators.**

Editing Codes	Effect
Z	Leading Zeros if any will be suppressed
*	Leading Zeros are replaced by asterisks(*)
$	Currency sign appears in the left most of the field.
-	Appears at left or right of the field as specified in the picture clause if value is negative
+	Appears if value is positive, else minus sign appears

- Editing Codes are specified in the picture clause for variables intended for report purpose.
- **These variables cannot be used for arithmetic calculations.**

Figure 2-17 EDITED Fields

More Editing Characters

EDIT CODES	MEANING
CR or DB	To be specified in the right most position of the pic clause. Appears only if the value is negative ,if value is positive it replaced by two characters.
.	Stands for decimal point. Cannot be specified with V clause
,	Inserted in the position where specified
B	Blank is appeared
0	Zero is appeared. To be specified left most position of pic clause.
-(hyphen) / (slash)	Used as date separators. Appears where specified.
BLANK WHEN ZERO	Sets all null values to blanks

Figure   2-18   EDITED Fields

EXAMPLES

DATA	PIC CLAUSE UNEDITED	PIC CLAUSE EDITED	EDITED VALUE
02346	9(5)	ZZ999	2346
0005	9(4)	ZZ99	05
03.42	99V99	Z999	003
0.007	9V999	ZV999	007
05634	9(5)	**999	*5634
00143	9(5)	$9(5)	$00143
453	9(3)	$**999	$**453
-0453	9(4)	-ZZ9(2)	-453
-0453	9(4)	9999-	0453-
453	9(3)	999-	453
-453	9(3)	999+	453-
70.46	99V99	99.99-	70.46
156758	9(6)	99/99/99	15/67/58
00	99V9	99.9 Blank when zero	
8654	9(4)	99b9b9	8654
24	99	9900	1200

Figure 2-19 EXAMPLES

Notes:

The above table shows contents of unedited fields in the first column. Contents of edited fields after moving the data-1 shown in last column.

Edited fields (Fields with editing codes) cannot take part in arithmetic computations. Moving of numeric edited fields to unedited fields is illegal.