

COBOL - DATA TYPES

http://www.tutorialspoint.com/cobol/cobol_data_types.htm

Copyright © tutorialspoint.com

Data Division is used to define the variables used in a program. To describe data in COBOL, one must understand the following terms:

- Data Name
- Level Number
- Picture Clause
- Value Clause

01	TOTAL-STUDENTS	PIC9(5)	VALUE '125'.
Level Number	Data Name	Picture Clause	Value Clause

Data Name

Data names must be defined in the Data Division before using them in the Procedure Division. They must have a user-defined name; reserved words cannot be used. Data names gives reference to the memory locations where actual data is stored. They can be elementary or group type.

Example

The following example shows valid and invalid data names:

Valid:

```
WS-NAME
TOTAL-STUDENTS
A100
100B
```

Invalid:

```
MOVE          (Reserved Words)
COMPUTE       (Reserved Words)
100           (No Alphabet)
100+B        (+ is not allowed)
```

Level Number

Level number is used to specify the level of data in a record. They are used to differentiate between elementary items and group items. Elementary items can be grouped together to create group items.

Level Number	Description
01	Record description entry
02 to 49	Group and Elementary items
66	Rename Clause items
77	Items which cannot be sub-divided
88	Condition name entry

- **Elementary items** cannot be divided further. Level number, Data name, Picture clause and Value clause *optional* are used to describe an elementary item.
- **Group items** consist of one or more elementary items. Level number, Data name, and Value clause *optional* are used to describe a group item. Group level number is always 01.

Example

The following example shows Group and Elementary items:

```
DATA DIVISION.
WORKING-STORAGE SECTION.
01 WS-NAME      PIC X(25).           ---> ELEMENTARY ITEM
01 WS-CLASS     PIC 9(2)  VALUE  '10'. ---> ELEMENTARY ITEM

01 WS-ADDRESS.   ---> GROUP ITEM
    05 WS-HOUSE-NUMBER PIC 9(3).      ---> ELEMENTARY ITEM
    05 WS-STREET      PIC X(15).      ---> ELEMENTARY ITEM
    05 WS-CITY         PIC X(15).      ---> ELEMENTARY ITEM
    05 WS-COUNTRY     PIC X(15) VALUE 'INDIA'. ---> ELEMENTARY ITEM
```

Picture Clause

Picture clause is used to define the following items:

- **Data type** can be numeric, alphabetic, or alphanumeric. Numeric type consists of only digits 0 to 9. Alphabetic type consists of letters A to Z and spaces. Alphanumeric type consists of digits, letters, and special characters.
- **Sign** can be used with numeric data. It can be either + or .
- **Decimal point position** can be used with numeric data. Assumed position is the position of decimal point and not included in the data.
- **Length** defines the number of bytes used by the data item.

Symbols used in a Picture clause:

Symbol	Description
9	Numeric
A	Alphabetic
X	Alphanumeric
V	Implicit Decimal
S	Sign
P	Assumed Decimal

Example

The following example shows the use of PIC clause:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.

DATA DIVISION.
    WORKING-STORAGE SECTION.
    01 WS-NUM1 PIC S9(3)V9(2).
    01 WS-NUM2 PIC PPP999.
    01 WS-NUM3 PIC S9(3)V9(2) VALUE -123.45.
    01 WS-NAME PIC A(6) VALUE 'ABCDEF'.
```

```

01 WS-ID PIC X(5) VALUE 'A121$'.

PROCEDURE DIVISION.
    DISPLAY "WS-NUM1 : "WS-NUM1.
    DISPLAY "WS-NUM2 : "WS-NUM2.
    DISPLAY "WS-NUM3 : "WS-NUM3.
    DISPLAY "WS-NAME : "WS-NAME.
    DISPLAY "WS-ID : "WS-ID.
STOP RUN.

```

JCL to execute the above COBOL program:

```

//SAMPLE JOB(TESTJCL,XXXXXX),CLASS=A,MSGCLASS=C
//STEP1 EXEC PGM=HELLO

```

When you compile and execute the above program, it produces the following result:

```

WS-NUM1 : +000.00
WS-NUM2 : .000000
WS-NUM3 : -123.45
WS-NAME : ABCDEF
WS-ID : A121$

```

Value Clause

Value clause is an optional clause which is used to initialize the data items. The values can be numeric literal, alphanumeric literal, or figurative constant. It can be used with both group and elementary items.

Example

The following example shows the use of VALUE clause:

```

IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.

DATA DIVISION.
    WORKING-STORAGE SECTION.
    01 WS-NUM1 PIC 99V9 VALUE IS 3.5.
    01 WS-NAME PIC A(6) VALUE 'ABCD'.
    01 WS-ID PIC 99 VALUE ZERO.

PROCEDURE DIVISION.
    DISPLAY "WS-NUM1 : "WS-NUM1.
    DISPLAY "WS-NAME : "WS-NAME.
    DISPLAY "WS-ID : "WS-ID.
STOP RUN.

```

JCL to execute the above COBOL program:

```

//SAMPLE JOB(TESTJCL,XXXXXX),CLASS=A,MSGCLASS=C
//STEP1 EXEC PGM=HELLO

```

When you compile and execute the above program, it produces the following result:

```

WS-NUM1 : 03.5
WS-NAME : ABCD
WS-ID : 00

```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js