

COBOL - BASIC SYNTAX

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

Copyright © tutorialspoint.com

Character Set

'Characters' are lowest in the hierarchy and they cannot be divided further. The COBOL Character Set includes 78 characters which are shown below:

Character	Description
A-Z	Alphabets <i>UpperCase</i>
a-z	Alphabets <i>LowerCase</i>
0-9	Numeric
	Space
+	Plus Sign
-	Minus Sign or Hyphen
*	Asterisk
/	Forward Slash
\$	Currency Sign
,	Comma
;	Semicolon
.	Decimal Point or Period
"	Quotation Marks
(Left Parenthesis
)	Right Parenthesis
>	Greater than
<	Less than
:	Colon
'	Apostrophe
=	Equal Sign

Coding Sheet

The source program of COBOL must be written in a format acceptable to the compilers. COBOL programs are written on COBOL coding sheets. There are 80 characters position on each line of a coding sheet.

Character positions are grouped into the following five fields:

Positions	Field	Description
1-6	Column Numbers	Reserved for line numbers.

7	Indicator	It can have Asterisk * indicating comments, Hyphen – indicating continuation and Slash / indicating form feed.
8-11	Area A	All COBOL divisions, sections, paragraphs and some special entries must begin in Area A.
12-72	Area B	All COBOL statements must begin in area B.
73-80	Identification Area	It can be used as needed by the programmer.

Example

The following example shows a COBOL coding sheet:

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. HELLO.
000250* THIS IS A COMMENT LINE
000300 PROCEDURE DIVISION.
000350 A000-FIRST-PARA.
000400         DISPLAY "Coding Sheet".
000500 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:

```
Coding Sheet
```

Character Strings

Character strings are formed by combining individual characters. A character string can be a

- Comment,
- Literal, or
- COBOL word.

All character strings must be ended with **separators**. A separator is used to separate character strings.

Frequently used separators : Space, Comma, Period, Apostrophe, Left/Right Parenthesis, and Quotation mark.

Comment

A comment is a character string that does not affect the execution of a program. It can be any combination of characters.

There are two types of comments:

Comment Line

Comment line can be written in any column. The compiler does not check a comment line for syntax and treats it for documentation.

Comment Entry

Comment entries are those that are included in the optional paragraphs of an Identification Division. They are written in Area B and programmers use it for reference.

The text highlighted in **Bold** are the commented entries in the following example:

```
000100 IDENTIFICATION DIVISION.                                000100
000150 PROGRAM-ID. HELLO.                                       000101
000200 AUTHOR. TUTORIALSPPOINT.                               000102
000250* THIS IS A COMMENT LINE                                000103
000300 PROCEDURE DIVISION.                                     000104
000350 A000-FIRST-PARA.                                        000105
000360/ First Para Begins - Documentation Purpose             000106
000400      DISPLAY "Comment line".                             000107
000500 STOP RUN.                                              000108
```

JCL to execute 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:

```
Comment Line
```

Literal

Literal is a constant that is directly hard coded in a program. In the following example, "Hello World" is a literal.

```
PROCEDURE DIVISION.
DISPLAY 'Hello World'.
```

There are two types of literals as discussed below:

Alphanumeric Literal

Alphanumeric Literals are enclosed in quotes or apostrophe. Length can be up to 160 characters. An apostrophe or a quote can be a part of a literal only if it is paired. Starting and ending of the literal should be same, either apostrophe or quote.

Example

The following example shows valid and invalid Alphanumeric Literals:

```
Valid:
'This is valid'
"This is valid"
'This isn't invalid'

Invalid:
'This is invalid"
'This isn't valid'
```

Numeric Literal

A Numeric Literal is a combination of digits from 0 to 9, +, -, or decimal point. Length can be up to 18 characters. Sign cannot be the rightmost character. Decimal point should not appear at the end.

Example

The following example shows valid and invalid Numeric Literals:

```
Valid:
100
+10.9
-1.9
```

Invalid:

1,00

10.

10.9-

COBOL Word

COBOL Word is a character string that can be a reserved word or a user-defined word. Length can be up to 30 characters.

User-Defined

User-defined words are used for naming files, data, records, paragraph names and sections. Alphabets, digits, and hyphens are allowed while forming user-defined words. You cannot use COBOL reserved words.

Reserved Words

Reserved words are predefined words in COBOL. Different types of reserved words that we use frequently are as follows:

- **Keywords** like ADD, ACCEPT, MOVE, etc.
- **Special characters** words like +, -, *, <, <=, etc
- **Figurative constants** are constant values like ZERO, SPACES, etc. All the constant values of figurative constants are mentioned in the following table:

Figurative Constants

Figurative Constants	Description
HIGH-VALUES	One or more characters which will be at the highest position in descending order.
LOW-VALUES	One or more characters have zeros in binary representation.
ZERO/ZEROES	One or more zero depending on the size of the variable.
SPACES	One or more spaces.
QUOTES	Single or double quotes.
ALL literal	Fills the data-item with Literal.

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