



# Basics of JAVA

## In this chapter, you will learn about :

- Difference Between Java and C++
- Comments and Documentation
- Declarations of Variables
- Variable Types
- Operators
- Assignment Statement
- Input and Output Streams
- Input Statement
- Output Statement
- Decision Making Statement
- Loop Statement
- While loop
- do-while loop
- Switch Statement

Java is a general purpose object oriented Programming (OOP) language developed by Sun Microsystems of USA in 1991.

Major features of JAVA are :

- It is transferable (i.e, can be moved from one computer to another).
- It can run with any operating system (OS).
- It is dynamic and extensible.

## DIFFERENCE BETWEEN JAVA AND C++

*You have already learned some basics of C++.*

*Java differs from C++ in following ways :*

1. Java does not support operator overloading whereas C++ does.
2. Java does not support multiple inheritance whereas C++ does
3. Java does not support global variable where as C++ does.
4. Java does not use pointers whereas C++ does.
5. Unlike C++, there are no header files in Java.

## COMMENTS AND DOCUMENTATION

There are two ways to indicate comments in a Java program. The first is two backslashes (//). This indicates the compiler to ignore the rest of the lines.

The second is to have /\* and \*/ surrounding the text that you wish to comment useful for multiline comments.

### **Character Set**

Set of different symbols available in a language is known as character set.

Various characters in Java are defined by unicode.



It is a 16-bit Character coding system and supports more than 34,000 Characters.

### **Keywords**

A keyword is a word that has a predefined meaning and this meaning cannot be changed. Java has 60 keywords. Some keywords are :

If, in, float, else, break, class, public, private keywords are also known as Reserve words.

### **Identifiers**

The names of variables, classes, and methods are identifiers. The rules for identifiers in Java are similar to most of the other languages.

They are listed as follows :

- Case-sensitive. (Capitalization Matters, i.e, 'Numbers' is not the same as 'number'.)
- In multiple words, each word start with an upper case letter, while the other letters are lower case.
- Class names start with a capital letter.
- Variable start with lower case letters.

### **Data Types**

Each variable in JAVA has a data type. Data type specifies the size and type of value that can be stored in the variable.

*Major data types in JAVA are :*

- 1. Integer Data Type :** It can store only whole numbers. Java supports four types of integers. These are byte, short, int and long.
- 2. Floating Data Type :** It can store whole number as well as fractional number. Java supports two type of floating data types. These are float and double.
- 3. Character Data Type :** It can store character values. Java provides a character data type called char.
- 4. Boolean Data Type :** It can store value True or False.

### **DECLARATIONS OF VARIABLES**

In Java, variables are names of storage locations. After assinging suitable name to a variable, you must specify the data type.

*Syntax / Format :*

data-type variable 1, variable 2, ..... variable n; where, the value of n is declared by the user.

**Example :** double price, cost;

### **VARIABLE TYPES**

*Mainly variables can be defined in two data types :*

- Primitive Type
- Non-primitive Type



The Primitive data types are for variables that contain a single piece of information, i. e; byte, short, int, long, float, double char, boolean.

Non-Primitive data types are also known as composite or reference data types. These data types are defined by the programmers as per their requirement. Mainly, these are used for objects of classes.

## OPERATORS

In previous chapter you have already studied, about different types of operators. Java also consists of arithmetic operators, relational operators, boolean logical operators, etc.

Some of the operators are as follows :

### Relational operators

Operators	Meaning
<	less than
>	greater than
<=	less than equal to
>=	greater than equal to
= =	equality
! =	not equal to

### Boolean operators

Operator & & means AND

Operator || means OR

Operator ! means NOT

## ASSIGNMENT STATEMENT

This statement is used to store a constant value to a variable. Variables are assigned value with the = sign.

**Syntax :** <data type > variable =value:

**Example :**

**First method :** int x;

x=3; // assign the value 3 to the variable x

**Second method :** int x= 3;

As you know, +, -, \*, and / operators are used for Mathematical operations.

**Example :** value 5/2 evaluates to 2 because the remain is eliminated. However, 5.0/2 evaluates to 2.5 because one of the operands is a real number.



For int, there is an additional operator, i.e. %. This is known as Modulus operator which returns the remainder after doing division.

**Example :**  $5\%2$  evaluates to 1, because 5 divided by 2 is 2 with a remainder of 1.

The increment operator (++) or decrement operator (-) can be used as integers.

## INPUT AND OUTPUT STREAMS

Java comes with predefined class libraries. All I/o (input / output) operations are defined in the Java. io package. It is always defined on the first line of the program.

**Example :** Import Java. io. \*; // the \* means to import everything that is a part of java. io Package.

## INPUT STATEMENT

*Java allows you to get input or read the variable using various methods. Some of them are listed below :*

### Method 1

```
Buffered Reader Stdin : new Buffered Reader [new Input Stream Reader (system. in)]
```

### Method 2

```
Input Stream Reader IR = new Input Stream Reader (system. in); Buffered Reader br = new Buffered Reader (IR);
```

*Here :*

Buffered Reader is class and can read a string of characters.

Stdin or br is a variable used for standard input.

Example : Program to read a line of text

Method 1

```
Import Java. io. *;
```

```
class text
```

```
{
```

```
Public static void Main (string args [ ] ) throws Io Exception // you will study later about this statement.
```


```
{
```

```
Buffered Reader stdin = new Buffered Reader [new Input stream Reader (system. in)]
```

```
String s; // variable S is declared for string.
```

```
System. out.println ("Enter text");
```

```
S = stdin.readline ( );
```



```
System.out.println ("A line of next is:" + S);
    }
}
```

## Method 2

```
import java.io.*;
class yashu
{
    public static void main (string args []) throws IO Exception.
    {
        Input Stream Reader IR = new Input Stream Reader (system.in);
        Buffered reader br = new Buffered Reader (IR);
        String nam; // nam is a variable for name.
        int rn; // rn is a variable for roll number.

        System.out.println ("Enter your name");
        nam = br.readLine ();
        System.out.println ("Enter your Roll No.")
        m = Integer.parseInt (br.readLine ( ));
        System.out.println ("My name is" + nam + "and Roll no. is" + m);
    }
}
```

## Program output :

```
Enter your name : Navit
Enter your Roll no : 999
My Name is Navit and Roll. no. is 999
System.out.println (" Enter the Price");
double price = Double.parseDouble (br.readLine ( ));
(Now, you may enter value in decimal.)
```

## OUTPUT STATEMENT

This statement is used to display the result on the screen. Output may be obtained by the help of the following statements.;

1. System.out.println ( );
2. System.out.print ( )'



**Example :**

```
System.out.print ("hello")  
System.out.Print in ("see you")  
System.out.Print in (later")
```

**Output :**

Hello see you  
later.

**Example 2**

```
int x = 10;  
system. out. print ("wel");  
system . out . print in ("come");  
would display :  
welcome  
whereas,  
int x =10 ;  
system. out. print ("wel);  
system. out. print in (x);  
would display:  
wel 10
```

You can also display multiple items to output by using the + operator to concatenate string.

**Example 3**

```
int x = 2 ;  
system . out. Print in ("welcome" + x + "India");  
would display ;  
welcome 2 India.
```

**DECISION MAKING STATEMENT**

This statement allows the user to take a decision depending on the given conditions.

Java has 'If' and 'else if' statements that operate like in other languages.

**Syntax / Format :**

- (i) if (condition)

**Statement 1;**

else



**Statement 2 ;**

If the condition is true, statement 1 is executed otherwise statement 2 is executed.

**Example :**

To find largest of two numbers.

If (a>b)

System.out. PrintIn (a);

else

System. out. PrintIn (b);

If you want more than one statement to be executed, you need to use the braces {—}.

Syntax / Format

(ii) If (condition)

{

*Statement 1 ;*

*Statement 2 ;*

}

**{ else**

**Statement 3 ;**

**Statement 4 }**

(iii) if (condition)

Statement;

Else if (condition)

Statement;

else if (condition)

statement;

**LOOP STATEMENT**

This statement is used to repeat set of statements.

The following loops are available;

- For loop
- while loop
- do-while loop

***For loop***

The for statement is used to execute a set of statement for a known number of times.



### Syntax / Format :

```
For (initialize statement; condition; increment statement)
{
Statement 1 ;
Statement 2 ;
.
.
.
Statement n ;
}
```

**Example :** For (in 1 = 0; (< 10; i = i+1)  
System. out. Print In (i);

### WHILE LOOP

The while statement performs a set of statements for a number of times, till the condition is satisfied.

### Syntax/ Format :

```
while (condition)
{
Statement 1 ;
Statement 2 ;
Statement n ;
}
```

If the condition is true, the body of the loop is executed, and then the condition is checked again. It is repeated until condition becomes false.

### DO-WHILE LOOP

These are similar to while Loops except that the loop body executes once before checking the condition. This means that first the body to loop is executed and then the condition is checked.

### Syntax : format :

```
Do
{
Statement 1 ;
Statement 2 ;
.
.
.
Statement n;
}
while (condition);
```





### Comparisons Example :

```
int x = 5;
while (x < 5)
{ System. out. Print In (x);
X++;
}
```

It will display nothing as a variable is not less than 5

```
int x = 5;
do
{ System. out. PrintIn (x);
  x++;
} while (x<5);
```

It will display number 5 before checking the condition.

### SWITCH STATEMENT

The switch statement is also known as user's choice statement OR menu driven statement. This statement allows you to make selective execution of the blocks (choice from a list of alternatives).

#### **Syntax/ Format;**

```
switch (expression)
{
case value 1;
statement;
break;
case value 2:
statements;
break;
.
.
.
case value n;
statements;
break;
default;
statement;
{
```



Example :

```
import java. io*;  
class menu  
{  
public static void main (string args [])  
{  
int a =10, b = 5, ch;  
Input Stream Reader IR = new Input Stream Reader (system. in);  
Buffered Reader br = new Buffered Reader (IR);  
System. out. PrintIn ("Menu");  
System. out. PrintIn ("1. For Addition");  
System. out. PrintIn ("2. For Subtraction")  
System. out. PrintIn ("3. For Multiplication");  
System. out. PrintIn ("4. for Exit")  
System. out. Print ("Enter your choice (1-4) ) ;  
Ch = Integer- Parse Int [br. read line ( )];  
Switch (choice)  
{  
Case 1;  
result = a+b;  
System. out. PrintIn ("Addition =" + result);  
break;  
Case 2;  
result = a - b;  
System. out. PrintIn (subtraction =" + result);  
break;  
Case 3;  
result = a*b;  
System. out. PrintIn ("Multiplication =" + result):  
break;  
default;  
System. out. PrintIn ("Exit");  
}  
}  
}
```



**Output :**

Menu

1. For Addition,
2. For Subtraction,
3. For Multiplication,
4. Exit

Enter your choice (1-4) ? =

Subtraction = 5

**Important terms :**

**Testing** : is dry run of the program.

**Debugging** : is removal of errors from a program.

**Syntax error** : is the grammatical or punctuation mistake made in the program

**Logical error** : is the wrong logic applied in the program.

**Run time error** : is due to any fault in computer component.

**Points to Remember**

- Java is a general purpose object oriented programming (OOP) language developed by Sun Microsystems of USA in 1991.
- There are two ways to indicate comments in a Java program.
- Set of different symbols available in a language is known as Charter set.
- Java has 60 keywords.
- The names of variables, classes & methods are Identifiers.
- Each variable in JAVA has a data type.
- Java comes with predefined class libraries.



**A. Tick (✓) the correct option :**

1. \_\_\_\_\_ indicates the computer to ignore the rest of the lines.  
 (a) //  (b) \* /  (c) / \*
2. Java has \_\_\_\_\_ keywords.  
 (a) 50  (b) 60  (c) 70
3. The \_\_\_\_\_ performs a set of statements for a number of times, till the condition is satisfied.  
 (a) While Loop  (b) Do-while Loop  (c) For Loop



**B. Fill in the blanks :**

1. \_\_\_\_\_ statement is used to display result on the screen.
2. Java has \_\_\_\_\_ & \_\_\_\_\_ statements that helps in decision making.
3. The \_\_\_\_\_ is used to execute a set of statements for a unknown number of homes.
4. The switch statement is also known as \_\_\_\_\_ or \_\_\_\_\_.
5. Java provides a character data types called \_\_\_\_\_.

**C. Write (T) for true and (F) for false :**

1. The names of variables, classes & methods are identifiers.
2. There are several header files in JAVA.
3. Java has 69 keywords.
4. Java comes with predefined class libraries.
5. The for statement is used to execute a set of statement for a known number of homes.

**D. Define the following :**

1. Comments
2. Identifiers
3. Character set
4. Parameters

**E. Answer the following questions :**

1. Differentiate between class variables and object variables.
2. What is a postfix expression? Explain with example.
3. State two differences & two similarities between 'if' & 'Switch Case'.
4. Explain various types of loops.
5. What do you mean by nesting of 'if' statement?