# Section 1- Basic Java

1. Variables & Data types
2. Keywords & Identifiers
3. Methods
4. Constructor
5. Control Statements
6. Loops

# Variables

* + Variables are nothing but **piece of memory** which is used to **store information**.
  + One variable can store **1 information** at a time.
  + Variables also used for **information reusability**.
  + To utilize variables in java programming language we need to follow below steps:

1. Variable declaration (Allocating/Reserving memory)
2. Variable Initialization (Assigning or inserting value)
3. Usage

**Note: -**

* + According to all programming language dealing with information directly is not a good practice to overcome this variables are introduced.

**Data Types-**

**Data Types:**

* Data types are used to represent **type of data or information** which we are going to use in our java program.
* In java programming it is mandatory to declare datatype before declaration of variable.
* In java, datatypes are classified into two types :

1. Primitive datatype.

         2. Non-primitive datatype.

**1.Primitive datatype:**

* There are **8 type** of primitive datatypes.
* all the primitive datatypes are **keywords.**
* **Memory size** of primitive datatype are **fix.**

The types of primitive datatype are:

**Note:-**

* keyword starts with lower case
* Primitive datatype starts with lower case

**syntax:**     datatype variablename;

1. **Numeric + Non-decimal:-**

Ex: 80,85,10,..etc

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **DataTypes** | **Size** | **Range** |
| 1 | byte | 1 byte | from -128 to 127 |
| 2 | short | 2 bytes | from -32,768 to 32,767 |
| 3 | int | 4 bytes | from -2,147,483,648 to 2,147,483,647 |
| 4 | long | 8 bytes | from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |

1. **Numeric + decimal:-**

Ex: 22.5, 22.8, 6.4....

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **DataTypes** | **Size** |
| 1 | float | 4 byte |
| 2 | double | 8 byte |

1. **Character :-**

  Ex: A,B,X,Z.

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **DataTypes** | **Size** |
| 1 | char | 2 byte |

1. **Conditional:**-

Ex:  true,false.

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **DataTypes** | **Size** |
| 1 | boolean | 1 bit |

2. Non-primitive datatype:

* There are **2 types** of non primitive datatypes .
* all the Non primitive datatypes are **identifiers**.
* **Memory** size of non primitive datatype is not defined or **not fix**.

Note: Identifier starts with capital letter.

Non-primitive datatype starts with capital letter.

e.g.  String, ClassName

**Methods:-**

A method is a block of code which only runs when it is called.

You can pass data, known **as parameters/arguments**, into a method.

Methods are used **to perform certain actions**, and they are also known as functions.

Why do we use methods? **To reuse code**: define the code once, and use it many times.

**1. Main method**

In any Java program, the main() method is the starting point from where JVM starts program execution.

So, the jvm needs to call the main() method.

Without main method we can't run any java program.

**2. Regular method**

1. **static regular method**

a) static method call from same class

//methodname();

b) static method call from diffrent/another class

//classname.methodname();

**2. non- static regular method**

c. non-static method call from same class

//classname objectname =new classname();

//objectname.methodname();

d. non-static method call from diffrent/another class

//classname objectname = new classname();

//objectname.methodname();

**Note:-**

At the time of program execution main method is going to get executed automatically, whereas regular methods are not going to get executed automatically.

At the time of program execution priority is scheduled for main method only.

To call a regular method we need to make call method call from main method, until unless if the method call is not made regular method will not get executed.

Regular methods can be called multiple times.

**5. method without/zero parameter**

**6. method with parameter.**

**7. method with return type**

**4. Control Statements-**

**1. if**

**2. if else**

**3. else if**

**4. nested if**

**5. switch**

**1. if**

if(conditon)

{

}

**2. if else**

**if** (condition)

{

}

**else**

{

}

**3. else if**

if (condition)

{

}

else if (condition)

{

}

else

{

}

**4. nested if**

if (condition)

{

if (condition)

{

}

else

{

}

}

else

{

}

**5. Switch**

**switch** (key)

{

**case** value: **break**;

**default**:**break**;

}

**5. loops**

**1. for loop**

**2. while**

**3. do while**

**1. for loop**

**Syntax for for loop-**

**for** (**int** i = 0; i < args.length; i++)

{

}

**2. while**

**Syntax for while loop-**

**while** (condition)

{

Syso (“ ”);

i++;

}

**3. do while**

**Syntax of do while loop**

**do**

{

Syso (“ ”);

i++;

}

**while** (condition);

**Types of variable:**

**1. local variable**

Creating **variable inside method/block** is known as local variable.

Scope of local variable **remains only within the method** & they are **temporary**.

**2. global variable**

Creating variable **outside method/block** is known as global variable.

Scope of global variable **remains throught the class** & they are **permanent**.

**3. class/static variable**

**Declaring** the variable using **static keyword** is known as class/static variable .

To access static variable from same class we need to make use of below statement:

**variablename;**

To access static variable from diff class we need to make use of below statement:

**classname.variablename;**

**4. Instance/non-static variable**

All the non-static variables are known as instance variable because to access non-static variable instance (object) need to be created.

To access non-static variable we need to make use of below statement:

**a) Create object of class.**

**b) objectname.variablename;**

**6. Constructor**

A constructor in Java is a special method that is used to initialize objects/variables.

The constructor is called when an object of a class is created.

At the time of constructor declaration below points need to follow:

1. Constructor name should be same as class name

2. You should not declare any return type for the constructor (like void).

3. any no of constructor can be declared in a java class but constructor name

should be same as class name,but arguments/parameter should be different.

Use of Constructor

1. To initialize data member/variable

2. To copy/load non-static members of class into object --> when we create

object of class

Types of Constructor

1. Default Constructor

2. User defined Constructor

**1. Default Constructor**

If Constructor is not declared in java class then at the time of compilation compiler will provide Constructor for the class

If programmer has declared the constructor in the class then compiler will not provide default Constructor.

The Constructor provided by compiler at the time of compilation is known as Default Constructor

**2. User defined Constructor**

If programmer is declaring constructor in java class then it is considered to be as User defined constructor.

User defined Constructor are classified into 2 types

1. Without/zero parameter constructor

2. With parameter constructor