Constructors in Java

- **What is a constructor?**

A constructor is a special method that is used to initialize an object. Every class has a constructor either implicitly or explicitly.

If we don't declare a constructor in the class then JVM builds a default constructor for that class. This is known as default constructor.

A constructor has same name as the class name in which it is declared. Constructor must have no explicit return type. Constructor in Java can not be abstract, static, final or synchronized. These modifiers are not allowed for constructor.

- **Syntax to declare constructor**

```
className (parameter-list){
  code-statements
}
```

className is the name of class, as constructor name is same as class name.

parameter-list is optional, because constructors can be parameterized and non-parameterized as well.

Constructor Example

In Java, constructor structurally looks like given in below program. A Car class has a constructor that provides values to instance variables.

```
class Car
{
        String name ;
        String model;
        Car( )    //Constructor
        {
                name ="";
                model="";
```

```
    }
}
```

Types of Constructor

Java Supports two types of constructors:

Default Constructor

Parameterized constructor

Each time a new object is created at least one constructor will be invoked.

```
Car c = new Car()      //Default constructor invoked
Car c = new Car(name); //Parameterized constructor invoked
```

Default Constructor

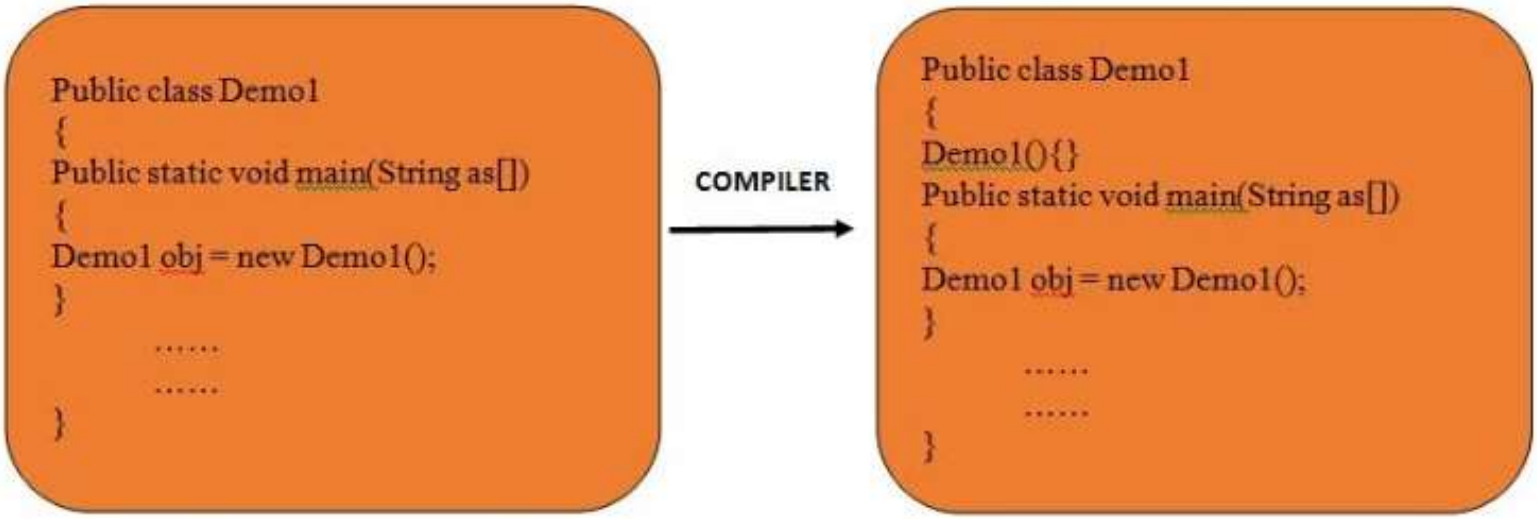
In Java, a constructor is said to be default constructor if it does not have any parameter. Default constructor can be either user defined or provided by JVM.

If a class does not contain any constructor, then during runtime JVM generates a default constructor which is known as system define default constructor.

If a class contain a constructor with no parameters then it is known as default constructor defined by user. In this case JVM does not create default constructor.

The purpose of creating constructor is to initialize states of an object.

The below image shows how JVM adds a constructor to the class during runtime.



- **User Define Default Constructor**

Constructor which is defined in the class by the programmer is known as user-defined default constructor.

Example:

In this example, we are creating a constructor that has same name as the class name.

```
class AddDemo1
{
        AddDemo1()
        {
                int a=10;
                int b=5;
                int c;
                c=a+b;
                System.out.println("*****Default Constructor*****");
                System.out.println("Total of 10 + 5 = "+c);
        }
        public static void main(String args[])
        {
```

```
        AddDemo1 obj=new AddDemo1();
    }
}
```

Constructor Overloading

Like methods, a constructor can also be overloaded. Overloaded constructors are differentiated on the basis of their type of parameters or number of parameters. Constructor overloading is not much different than method overloading. In case of method overloading, you have multiple methods with same name but different signature, whereas in Constructor overloading you have multiple constructors with different signature but only difference is that constructor doesn't have return type.

Example of constructor overloading

```
class Cricketer
{
    String name;
    String team;
    int age;
    Cricketer ()   //default constructor.
    {
        name ="";
        team ="";
        age = 0;
    }
     Cricketer(String n, String t, int a)   //constructor overloaded
    {
        name = n;
        team = t;
        age = a;
    }
```

```
	Cricketer (Cricketer ckt)  //this is similar to copy constructor of c++
	{
		String s="HI";
		name = ckt.name+s;
		team = ckt.team+s;
		age = ckt.age;
	}
	void printit()
	{
		System.out.println("this is " + name + " of "+team);
	}
}
class test
{
	public static void main (String[] args)
	{
		Cricketer c1 = new Cricketer();
		Cricketer c2 = new Cricketer("sachin", "India", 32);
		Cricketer c3 = new Cricketer(c2 );
		c1.printit();
		c2.printit();
		c1.name = "Virat";
		c1.team= "India";
		c1.age = 32;
		c3.printit();
	}
}
```