Constructor

We will cover the following

- What is a constructor?
- Default constructor
- Parameterized constructor
- Java Copy Constructor
- Diff b/w constructor and method

What is a Constructor?

It is a special method that is used to initialize the object when an object of a class is created in the program. As the name suggests, the constructor is used to construct the object of a class. It is called when an instance of the class is created.

- A constructor's name must be exactly the same as the name of its class.
- The constructor is a special method because it does not have a return type. We do not even need to write void as the return type.
- The purpose of a Java constructor is to initialize the newly created object before it is used.
- Every time an object is created using the new() keyword, at least one constructor is called.

There are two types of the constructors

- Default constructor
- Parameterized constructor

Default constructor

A constructor is called a "Default Constructor" when it doesn't have any parameter.

Main .java

```java
public class Main
{
    public static void main(String[] args) {
        // we are creating the object of the car class
        car c=new car();
    }
}

// car class
class car{

   car(){System.out.println("car is created");}

}
```

```
car is created


...Program finished with exit code 0
```

Here you can notice that we have created the object of the car class and the car constructor of the car class is called.

Point to be remembered: If there isn't any constructor in a class, the compiler automatically creates a default constructor within the class that will not be visible to us.

Let us assume we have a bike class and we have not created any constructor within the bike class then in that situation compiler will create the bike() constructor automatically that will not be visible in the code.

Main.java

```
 9  public class Main
10  {
11      public static void main(String[] args) {
12          // we are creating the object of the Bike class
13          Bike b=new Bike();
14      }
15  }
16
17  // Bike class
18  class Bike{
19
20
21  }
22
23
```

```
...Program finished with exit code 0
```

Here you can notice that the program is still running as a default constructor will be created by the compiler automatically within the bike class which is not visible in the program.

Purpose of a default constructor:-

Basically, the purpose of the default constructor is to provide the default values to the objects like null, 0, etc. according to the type.

```
 9  public class Main
10  {
11      public static void main(String[] args) {
12          //creating objects
13          Student s1=new Student();
14          Student s2=new Student();
15          //displaying values of the object
16          s1.display();
17          s2.display();
18      }
19  }
20
21  class Student{
22      int id;
23      String name;
24      //method to display the value of id and name
25      void display()
26      {
27          System.out.println(id+" "+name);
28      }
29
```

```
0 null
0 null

...Program finished with exit code 0
```

Parameterized constructor:-

A constructor which has a certain number of parameters is called a parameterized constructor.

Purpose of a parameterized constructor:- The parameterized constructor is used to initialize the object with different-different values.

```
 9  public class Main
10  {
11      public static void main(String[] args) {
12          //creating objects
13          Student s1=new Student(12,"shacksham");
14          Student s2=new Student(14,"Deepak");
15          //displaying values of the object
16          s1.display();
17          s2.display();
18      }
19  }
20
21  class Student{
22      int id;
23      String name;
24
25      Student(int stdid,String stdname) {
26          id=stdid;
27          name=stdname;
28      }
29
30      void display(){
31          System.out.println(id+"  "+name);
32      }
33
34  }
```

```
12  shacksham
14  Deepak


...Program finished with exit code 0
```

Java Copy Constructor

There is no copy constructor in the Java language. But, we can copy the values from one object to another object like copy constructor in other programming languages.
There are several ways to copy the values of one object into another object in Java. They are:

- By constructor
- By assigning the values of one object to another
- By clone() method of Object class

By constructor

Here we are copying the values from one object to the other using the copy constructor CopyConstructroExample. You can see in the following code

.

Main.java

```java
class CopyConstructorExample {
    private double x, y;
    // A normal parameterized constructor
    public CopyConstructorExample(double one, double two) {
        x = one;
        y = two;
    }
    // copy constructor
    CopyConstructorExample(CopyConstructorExample c) {
        System.out.println("Copy constructor called");
        x = c.x;
        y = c.y;
    }
    public String display() {
        return "(" + x + " " + y + "i)";
    }
}

public class Main {

    public static void main(String[] args) {
        CopyConstructorExample c1 = new CopyConstructorExample(10, 15);

        // Following involves a copy constructor call
        CopyConstructorExample c2 = new CopyConstructorExample(c1);

        // Note that following doesn't involve a copy constructor call as
        // non-primitive variables are just references.
        CopyConstructorExample c3 = c2;

        System.out.println(c2.display()); // display() of c2 is called here
    }
}
```

input

```
Copy constructor called
(10.0 + 15.0i)


...Program finished with exit code 0
```

By assigning the values of one object to another

Here we are copying the values from one object to the other By assigning the values of one object to another. You can see in the following code

Main.java

```
class CopyConstructorExample {
    double x, y;
    // A normal parameterized constructor
    public CopyConstructorExample(double one, double two) {
        x = one;
        y = two;
    }
    //default constructor
     public CopyConstructorExample (){

     }

    public String display() {
        return "(" + x + " + " + y + "i)";
    }
}

public class Main {

    public static void main(String[] args) {
        CopyConstructorExample c2 = new CopyConstructorExample(10, 15);


        CopyConstructorExample c3 =new CopyConstructorExample();
        //here we are copying values of one constructor into another.
        c3.x=c2.x;
        c3.y=c2.y;

        System.out.println(c3.display()); // display() of c3 is called here
    }
}

```

input

```
(10.0 + 15.0i)


...Program finished with exit code 0
```

Diff b/w Constructor and method in java

You need to understand that constructor is different than the method in various ways:-