

# CONSTRUCTORS

## Java Object Oriented Programming

Author:- Vishal Pandey

### ***Constructor in Java:-***

- Constructors are similar to methods, but they are used to initialize an object.
- Constructors do not have any return type(not even void).
- We can create an object by using the new() keyword, a constructor is called.
- If we don't create a constructor by ourself, then the default constructor is called by the Java Compiler.

### ***Rules for creating a constructor:-***

- The name of the class and the constructor should be same.
- It must have no explicit return type.
- It cannot be abstract, static, final and synchronized.

### ***Types of Constructor:-***

There are two types of constructor in Java:-

- Default Constructor
- Parameterized Constructor.

1. **Default Constructor:-** *A constructor with zero(0) parameters is known as default constructor.*

eg:-

```
<class_name>(){  
    //code to be executed  
}
```

eg:-

```
class constructor{
    constructor(){
        System.out.println("This is default constructor");
    }
}

public class default{
    public static void main(String[] args){
        constructor c = new constructor();
    }
}
```

**2. Parameterized Constructor:-** *A constructor with some specified number of parameters is known as a parameterized constructor.*

eg:-

```
<class_name>(<data-type> param1, <data-type> param2,...){
    //code to be executed
}
```

eg:-

```
class constructor{
    constructor(int a, String s){
        System.out.println("This is " + a + "of " + s);
    }
}

public class default{
    public static void main(String[] args){
        constructor c = new constructor("Vishal", 20);
    }
}
```

### ***Constructor Overloading in Java:-***

*Just like the methods, we can overload the constructor in Java.*

*Note:-*

1. *Constructor can take parameters without being overloaded.*

2. *There can be more than two overloaded constructors.*

eg:-

```
class Employee {
// First constructor
    Employee(String s, int i){
        System.out.println("The name of the first employee is : " + s);
        System.out.println("The id of the first employee is : " + i);
    }
// Constructor overloaded
    Employee(String s, int i, int salary){
        System.out.println("The name of the second employee is : " + s);
        System.out.println("The id of the second employee is : " + i);
        System.out.println("The salary of second employee is : " + salary);
    }
}

public class constructors {
    public static void main(String[] args) {
        Employee vishal = new Employee("Vishal",1);
        Employee bill = new Employee("Bill",2,70000);
    }
}
```