Constructors in Java

In [Java](https://www.javatpoint.com/java-tutorial), a constructor is a block of codes similar to the method. It is called when an instance of the [class](https://www.javatpoint.com/object-and-class-in-java) is created. At the time of calling constructor, memory for the object is allocated in the memory.

It is a special type of method which is used to initialize the object.

Every time an object is created using the new() keyword, at least one constructor is called.

It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.

There are two types of constructors in Java: no-arg constructor, and parameterized constructor.

**Note:** It is called constructor because it constructs the values at the time of object creation. It is not necessary to write a constructor for a class. It is because java compiler creates a default constructor if your class doesn't have any.

### **Rules for creating Java constructor**

There are two rules defined for the constructor.

1. Constructor name must be the same as its class name
2. A Constructor must have no explicit return type
3. A Java constructor cannot be abstract, static, final, and synchronized

#### Note: We can use [access modifiers](https://www.javatpoint.com/access-modifiers) while declaring a constructor. It controls the object creation. In other words, we can have private, protected, public or default constructor in Java.

## **Types of Java constructors**

There are two types of constructors in Java:

1. Default constructor (no-arg constructor)
2. Parameterized constructor



## **Java Default Constructor**

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

### **Syntax of default constructor:**

<class\_name>()

{

}

## **Example of default constructor**

|  |
| --- |
| In this example, we are creating the no-arg constructor in the Bike class. It will be invoked at the time of object creation. |

//Java Program to create and call a default constructor

**class** Student {

//creating a default constructor

Student () {

System.out.println ("Welcome to KKWIEER");

}

//main method

**public** **static** **void** main(String args[]) {

//calling a default constructor

Student s = **new** Student();

}

}

Output:

Welcome to KKWIEER

#### **Rule: If there is no constructor in a class, compiler automatically creates a default constructor.**

### **Java Parameterized Constructor**

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

### **Why use the parameterized constructor?**

The parameterized constructor is used to provide different values to distinct objects. However, you can provide the same values also.

### **Example of parameterized constructor**

In this example, we have created the constructor of Student class that have two parameters. We can have any number of parameters in the constructor.

//Java Program to demonstrate the use of the parameterized constructor.

**class** Student{

**int** id;

    String name;

    //creating a parameterized constructor

    Student(**int** i , String n){

    id = i ;

     name = n ;

    }

    //method to display the values

**void** display(){

System.out.println(id+" "+name);

}

**public** **static** **void** main(String args[]){

    //creating objects and passing values

    Student s1 = **new** Student(111,"Sneha");

    Student s2 = **new** Student(222,"Mayuri");

    //calling method to display the values of object

    s1.display();

    s2.display();

   }

}

Output:

111 Sneha

222 Mayuri