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

## **Types of Java constructors**

Java Default Constructor

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

Syntax of default constructor

1. <class\_name>(){}
2. **class** Bike1{
3. //creating a default constructor
4. Bike1(){System.out.println("Bike is created");}
5. //main method
6. **public** **static** **void** main(String args[]){
7. //calling a default constructor
8. Bike1 b=**new** Bike1();
9. }
10. }

### **What is the purpose of a default constructor?**

The default constructor is used to provide the default values to the object like 0, null, etc., depending on the type.

### **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

## **Constructor Overloading in Java**

In Java, a constructor is just like a method but without return type. It can also be overloaded like Java methods.

Constructor [overloading in Java](https://www.javatpoint.com/method-overloading-in-java) is a technique of having more than one constructor with different parameter lists. They are arranged in a way that each constructor performs a different task. They are differentiated by the compiler by the number of parameters in the list and their types.

## **Difference between constructor and method in Java**

|  |  |
| --- | --- |
| **Java Constructor** | **Java Method** |
| A constructor is used to initialize the state of an object. | A method is used to expose the behavior of an object. |
| A constructor must not have a return type. | A method must have a return type. |
| The constructor is invoked implicitly. | The method is invoked explicitly. |
| The Java compiler provides a default constructor if you don't have any constructor in a class. | The method is not provided by the compiler in any case. |
| The constructor name must be same as the class name. | The method name may or may not be same as the class name. |

### **Does constructor return any value?**

Yes, it is the current class instance (You cannot use return type yet it returns a value).

### **Can constructor perform other tasks instead of initialization?**

Yes, like object creation, starting a thread, calling a method, etc. You can perform any operation in the constructor as you perform in the method.

### **Can we have a constructor in the Interface?**

No, we cannot have constructors in the Java interface.

### **Explain Constructor Chaining?**

Constructor Chaining is a way to call one constructor from another constructor with respect to the current object

### **What happens if we provide a return type to a constructor?**

If we provide a return type to a constructor, it will function as a general method. But, the compiler will display a warning message, "**This method has a Constructor name**".

### **What is a private constructor?**

Like methods, we can have the private constructors in Java. To make or create a constructor as private, use the **private** keyword while declaring it. It can only be accessed within that class.

The following are some usage scenarios when we need a private constructor:

* Internal Constructor chaining

### **Why constructors in Java cannot be static?**

The constructors cannot be static in Java. When we declare a method as static, it means the method belongs to the class and not to a specific object.

### **Why constructor name is similar to the class name?**

When we create an object of a class using a new keyword, it should have information about that particular class. That is why the constructor's name must be similar to the class name.

### **Why return type is not allowed for the constructor?**

The return type is not allowed in the constructor because if we provide a return type in the constructor, it will act as the normal method. So, to differentiate between constructor and method block, the return type is not allowed in constructors.