

# Java - Constructors

A constructor initializes an object when it is created. It has the same name as its class and is syntactically similar to a method. However, constructors have no explicit return type.

Typically, you will use a constructor to give initial values to the instance variables defined by the class, or to perform any other start-up procedures required to create a fully formed object.

All classes have constructors, whether you define one or not, because Java automatically provides a default constructor that initializes all member variables to zero. However, once you define your own constructor, the default constructor is no longer used.

## Syntax

Following is the syntax of a constructor –

```
class ClassName {  
    ClassName() {  
    }  
}
```

Java allows two types of constructors namely –

- No argument Constructors
- Parameterized Constructors

## No argument Constructors

As the name specifies the no argument constructors of Java does not accept any parameters instead, using these constructors the instance variables of a method will be initialized with fixed values for all objects.

Example

```
Public class MyClass {  
    Int num;  
    MyClass() {  
        num = 100;  
    }  
}
```

You would call constructor to initialize objects as follows

```
public class ConsDemo {
```

```

public static void main(String args[]) {
    MyClass t1 = new MyClass();
    MyClass t2 = new MyClass();
    System.out.println(t1.num + " " + t2.num);
}
}

```

This would produce the following result

100 100

## Parameterized Constructors

Most often, you will need a constructor that accepts one or more parameters. Parameters are added to a constructor in the same way that they are added to a method, just declare them inside the parentheses after the constructor's name.

Example

Here is a simple example that uses a constructor –

```

// A simple constructor.
class MyClass {
    int x;

    // Following is the constructor
    MyClass(int i) {
        x = i;
    }
}

```

You would call constructor to initialize objects as follows –

```

public class ConsDemo {
    public static void main(String args[]) {
        MyClass t1 = new MyClass( 10 );
        MyClass t2 = new MyClass( 20 );
        System.out.println(t1.x + " " + t2.x);
    }
}

```

This would produce the following result –

10 20

## References

- [www.tutorialspoint.com/java](http://www.tutorialspoint.com/java)

