

Website

<https://www.curiousinfotecholutions.com>

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Constructors in Java

[Video Link](https://youtu.be/xllkfdJQSHo)

<https://youtu.be/xllkfdJQSHo>

Java Programming Day-10



Email id:
info@curiousinfotecholutions.com

Syllabus Day-10

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Constructor

- ✓ **Introduction** of Java Constructor
- ✓ How Constructors are **Different** From Methods in Java?
- ✓ **Rules** for creating Java constructor
- ✓ **Syntax** of Java Constructor
- ✓ **Types** of constructors
- ✓ **No argument** Constructors
- ✓ **Parameterized** Constructors
- ✓ **Copy** Constructors in Java
- ✓ Constructor **Overloading** in Java



Introduction of Java Constructor



- It is **recommended** that all objects that are created must be given initial values.
- Already, we have done..
- The first approach uses the **dot operator** to access the instance variables and then values to them individually. However this is **tedious approach** to initialize all the variables of all the objects.
- The second approach takes the help of a method like **getData** to initialize each object individually using statements like:
 `object.getData(15,10);`

Contd..

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- A constructor in Java is a special method that is used to initialize objects.
- The constructor is called when an object of a class is created.
- It can be used to set initial values for object attributes.
- In Java, a constructor is a block of codes similar to the method.
- It is called when an instance of the class is created.
- At the time of calling the constructor, memory for the object is allocated in the memory.
- Every time an object is created using the new() keyword, ***at least one constructor is called.***

How Constructors are Different From Methods in Java?

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- Constructors must have the same name as the class within which it is defined while it is not necessary for the method in Java.
- Constructors do not return any type *while method(s) have the return type or **void** if does not return any value.*
- Constructors are called only once at the time of Object creation *while method(s) can be called any number of times.*

Rules for creating Java constructor



There are **three 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.*

Syntax of Java Constructor



- Following is the **syntax** of a constructor –

- class **ClassName** {
- **ClassName()** {
- }
- }

- Following is the example of a constructor –

```
class Rectangle
{
    int length, width;
    Rectangle()
    {
        length = 0; width = 0;
    }
}
```

Types of constructors

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Java allows two types of constructors namely –

1. No argument Constructors
2. Parameterized Constructors

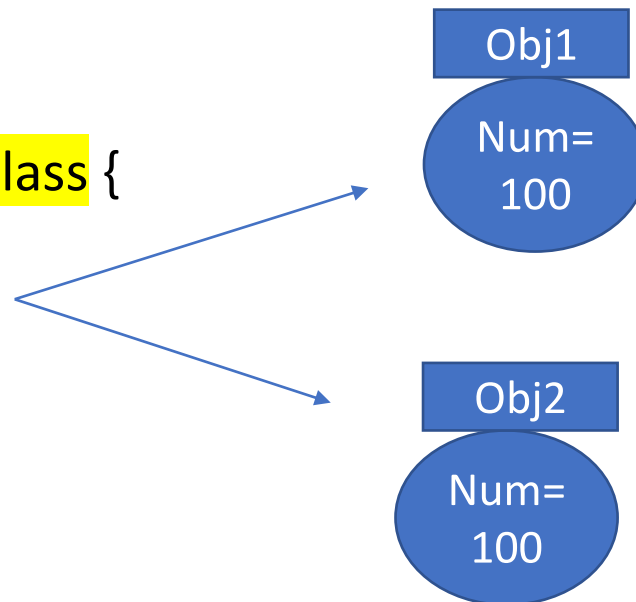
No argument Constructors



- 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;  
    }  
}
```



Example: No argument Constructors



```
• public class MyClass {  
•     int num;  
•     MyClass() {  
•         num = 100;  
•     }  
• }
```

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

Output
100 100

Parameterized Constructors

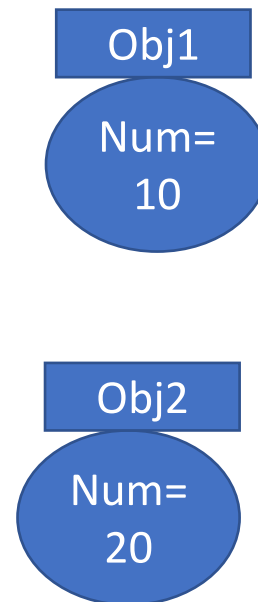
Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- Parameterized Constructors are 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

```
public class MyClass {  
    Int num;  
    MyClass(int x) {  
        num = x;  
    }  
}
```



Example: Parameterized Constructors



```
• class MyClass
• {
•     int number;
•     MyClass(int X)
•     {
•         number =X;
•     }
• }
```

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

Output
10 20

Copy Constructors in Java



```
• class Number {  
•     private int N;  
•     // A normal parameterized constructor  
•     public Number(int M) {N = M;}  
•     // copy constructor  
•     Number(Number A) {  
•         System.out.println("Copy constructor called");  
•         N = A.N;  
•     }  
•     void display(){System.out.println(N);}  
• }
```

```
• public class Main {  
•  
•     public static void main(String[] args) {  
•         Number N1 = new Number(10);  
•         N1.display();//10  
•  
•         Number N2 = new Number(N1);  
•         N2.display();//10  
•  
•         Number N3 = N2;  
•         N3.display();//10  
•     }  
• }
```

Constructor Overloading in Java



```
• class Add
• {
•   Add(int x, int y)
•   {
•       System.out.println(x+y);
•   }
•   Add(int x, int y, int z)
•   {
•       System.out.println(x+y+z);
•   }
• }
```

```
class Main{
    public static void main(String args[])
    {
        Add Obj1 = new Add(2,3);
        Add Obj2 = new Add(2,3,4);
    }
}
```