SNS COLLEGE OF TECHNOLOGY

Coimbatore-35.
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

COURSE CODE & NAME : 23CST205 - Object Oriented Programming Using Java

II YEAR/ III SEMESTER

UNIT – IV INHERITANCE AND POLYMORPHISM

Topic: Constructor

Constructors in Java

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.

How Java Constructors are Different From Java Methods?

- Constructors must have the same name as the class within which it is defined 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.

Need of Constructors in Java

Think of a Box. If we talk about a box class then it will have some class variables (say length, breadth, and height). But when it comes to creating its object(i.e Box will now exist in the computer's memory), then can a box be there with no value defined for its dimensions?

The answer is N**o**.
So constructors are used to assign values to the class variables at the time of object creation, either explicitly done by the programmer or by Java itself (default constructor).

When Java Constructor is called?

- Each time an object is created using a **new()** keyword, at least one constructor (it could be the default constructor) is invoked to assign initial values to the **data members** of the same class. Rules for writing constructors are as follows:
- The constructor(s) of a class must have the same name as the class name in which it resides.
- A constructor in Java can not be abstract, final, static, or Synchronized.
- Access modifiers can be used in constructor declaration to control its access i.e which other class can call the constructor.

Types of Constructors in Java

There are three types of constructors in Java are mentioned below:

- Default Constructor
- Parameterized Constructor
- Copy Constructor

1. Default Constructor in Java

- A constructor that has no parameters is known as default constructor. A default constructor is invisible. And if we write a constructor with no arguments, the compiler does not create a default constructor. It is taken out. It is being overloaded and called a parameterized constructor. The default constructor changed into the parameterized constructor. But Parameterized constructor can't change the default constructor. The default constructor can be implicit or explicit.

Implicit Default Constructor:

- If no constructor is defined in a class, the Java compiler automatically provides a default constructor. This constructor doesn't take any parameters and initializes the object with default values, such as 0 for numbers, null for objects.

Explicit Default Constructor:

If we define a constructor that takes no parameters, it's called an explicit default constructor. This constructor replaces the one the compiler would normally create automatically. Once you define any constructor (with or without parameters), the compiler no longer provides the default constructor for you.

```
// Default Constructor
import java.io.*;
// Driver class
class GFG {
    // Default Constructor
    GFG() { System.out.println("Default constructor"); }
    // Driver function
    public static void main(String[] args)
    {
        GFG hello = new GFG();
    }
}
```

OUTPUT : Default constructor

2. Parameterized Constructor in Java

- A constructor that has parameters is known as parameterized constructor. If we want to initialize fields of the class with our own values, then use a parameterized constructor.

```
import java.io.*;
class Geek {
    // data members of the class.
    String name; int id;
      Geek(String name, int id) {
        this.name = name;
        this.id = id;
    }}
class GFG
{ public static void main(String[] args)
    {
        // This would invoke the parameterized constructor.
        Geek geek1 = new Geek("Avinash", 68);
        System.out.println("GeekName :" + geek1.name
                           + " and GeekId :" + geek1.id);
    }}
```

OUTPUT:

GeekName :Avinash and GeekId :68

3. Copy Constructor in Java

- Unlike other constructors copy constructor is passed with another object which copies the data available from the passed object to the newly created object.
- we can copy the values from one object to another like copy constructor in C++.
- There are the following ways to copy the values of one object into another:
- By Using Constructor
- By Assigning the Values of One Object into Another
- By Using clone() Method of the Object Class


THANK
YOU!