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

COURSE NAME : 23CSB101 & Object Oriented Programming

I YEAR/ II SEMESTER

UNIT – I INTRODUCTION TO OOP & JAVA

Topic: Constructor

Dr.P.Poonkodi

Assistant Professor(SG)

Department of Computer Science and Technology

Constructor

- A **constructor** is a special method in Java that initializes objects
- 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
  - Constructor **name must be the same as its class name**
  - A Constructor must have **no explicit return type**
  - A constructor **cannot be abstract, static, final, and synchronized**

Types of Constructor

- Default Constructor
- Parameterized Constructor
- Copy Constructor
- **Syntax**

```
class Demo
{
  .......
  // A Constructor method
  Demo() {
  }
  .......
}
```

Default Constructor

- A constructor that has no parameters
- A default constructor is invisible
- if we write a constructor with no arguments, the compiler does not create a default constructor
- assigns default values
- default constructor changed into the parameterized constructor
- But Parameterized constructor can’t changed into default constructor

Default Constructor

```
class Box {
    // Default Constructor
    Box()
        {
            System.out.println("Default constructor");
        }
        public static void main(String[] args)
        {
            Box hello = new Box();
        }
}
```

Default Constructor

```
class Student
{
        String name;
        int age;
 // Default Constructor
        Student()
        {
        name = "Unknown";
        age = 0;
        }
```

```
void display()
{
System.out.println("Name: " + name + ",
Age: " + age);
}
public static void main(String[] args)
{
Student s1 = new Student();
// Constructor is called automatically
s1.display();
}
}
```

Parameterized Constructor

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

Parameterized Constructor

```
class Box
{
double width;
double height;
double depth;
```

// Default Constructor

```
Box()
{
width = 5;
height = 4;
depth = 3;
}
```

// Parameterized Constructor

```
Box(double w, double h, double d)
{
width = w;
height = h;
depth = d;
}
//Normal Method
double volume()
{
return width * height * depth;
}
}
```

Parameterized Constructor

```
class BoxDemo
{
    public static void main(String args[])
    {
        Box b1 = new Box();
        Box b2 = new Box(3, 6, 9);
        double vol;
        vol = b1.volume();
            System.out.println("Volume of box1 is " + vol);
        vol = b2.volume();
            System.out.println("Volume of box2 is " + vol);
     }
}
```

Copy Constructor

- a constructor that creates an object using another object of the same Java class

```
public class Student {
   private String name;
   private int age;
   // defining parameterized constructor
   public Student(String name, int age)
{
      this.name = name;
      this.age = age;
   }
```

Copy Constructor

```
// defining copy constructor
  public Student(Student s){
     this.name = s.name;
     this.age = s.age;
  }
  // creating display function
  public void display(){
     System.out.println("Name : "+this.name);
     System.out.println("Age : "+this.age);
  }
```

Copy Constructor

```
public static void main(String[] args){
    // create a new object of class Person
    Student s = new Student("Mohan", 15);
    // display original student details
    System.out.println("Displaying the original object");
    s.display();
    //  Display copied student details
    System.out.println("Displaying the copied object");
    Student copiedStudent = new Student(s);
    copiedStudent.display();
  }
}
```

References

- Java : the complete Reference (Eleventh Edition), Herbert Schildt, 2018.


Thank You!