Programming with Python and Java (CS 29008)

School of Electronics Engineering, KIIT DU

Chapter 4

Lab 4: Constructors in Java

Objective

To understand and apply the concepts of constructors in Java program.

Outcome

After completing this lab, the students would be able to develop Java programs using class, objects, methods and constructors.

4.1 What is a Constructor?

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 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.

4.2 Rules for creating a constructor

There are three rules associated with the constructor declaration. They are:

1. Constructor name must be the same as its class name.
2. A Constructor has no return type.
3. A Java constructor cannot be **abstract**, **static**, **final**, or **synchronized**.

4.3 Types of constructor in Java

There are two types of constructors in Java:

1. Default constructor

2. Parameterized constructor

4.3.1 Default constructor

A constructor is called **default constructor** when it does not have any parameter or argument. The syntax is

```
classname() {
   // constructor body
}
```

Example

The example shows a Java program without any constructor.

Listing 4.1: Java program without a constructor

```
class Rectangle {
        int height;
        int width;
        void display() {
        System.out.println("Width:" + width);
        System.out.println("Height:" + height);
        }
}
class Dctor1 {
        public static void main(String args[]) {
                // creating an object
                Rectangle rec = new Rectangle();
                rec.display();
        }
}
```

In the above example, the class **Rectangle** is defined with two member variables, **height** and **width** of type **int**. The member method **display()** prints the member data of the objects of the class **Rectangle**. Note that, we have not included any default constructor in the program. However, the Java compiler invokes the default constructor (even if it is not present in the program) during the object creation (i.e. at the time of executing **Rectangle rec = new Rectangle();** to initialize the member variables **height** and **width** to 0. If we execute the above code, we get the following output on the console:

Width: 0
Height: 0

We obtain the same output if a default constructor is added to the program as shown below.

Listing 4.2: Java program with a default constructor

```
class Rectangle {
        int height;
        int width;
    Rectangle() {
       height = 0; width = 0;
    }
        void display() {
        System.out.println("Width:" + width);
        System.out.println("Height:" + height);
        }
}
class Dctor1 {
        public static void main(String args[]) {
                // creating an object
                Rectangle rec = new Rectangle();
                rec.display();
        }
}
```

If we want to initialize the member variables to values other than 0, say, **height = 5** and **width = 10**, then default constructor or constructor with no arguments must be explicitly mentioned in the program as shown below.

```
Rectangle() {
       height = 5; width = 10;
    }
```

The execution of the Java program with the above default constructor displays output

Width: 10
Height: 5

4.3.2 Parameterized constructor

Constructors with parameters or arguments will have the following form:

```
classname(arguments) {
   // constructor body
}
```

The syntax of using the constructor for creating an object is as follows:

```
classname objectName = new classname(arguments);
```

Example

The following Java program shows the usage of a parameterized constructor.

```
class Rectangle {
        int height, width;
        Rectangle(int h, int w) {
                height = h; width = w;
        }
        int area() {
                return height * width;
        }
        void display() {
        System.out.println("Height=" + height);
        System.out.println("Width=" + width);
        }
}
class Pctor {
        public static void main(String args[]) {
        // creating object
        Rectangle rec = new Rectangle(5, 10);
        rec.display();
        System.out.println("Area=" + rec.area());
        }
}
```

In this example, there is a constructor with two arguments **h** and **w**. These two arguments set the values of the member variables **height** and **width** of the class **Rectangle**. When the Java compiler executes the line **Rectangle rec = new Rectangle(5, 10);** the parameterized constructor is invoked and the member variables **height** and **width** are set to 5 and 10 respectively. The execution of the entire program prints the following output on the command prompt:

```
Width = 10
Height = 5
Area = 50
```

4.4 Lab 4 Exercises

1. Create a class by name **Triangle** with the three sides **a**, **b**, and **c** as its member data. Include constructors and member methods to perform the following:

1. to accept the sides of a triangle.
2. to display the sides of a triangle.
3. to find whether the triangle is an equilateral or an isosceles or right angled triangle.

2. A complex number is of the form $x + iy$ where x is the real part and y is an imaginary part of the number. Design a Java class called **Complex** representing the complex number with member data x and y of the number. Include constructors and member methods to perform the following:

1. to accept and display a complex number
2. to find the sum of two complex numbers
3. to find the product of two complex numbers

3. Create a Java class called **Account** with the member data **account_number**, **name**, **balance**. Using constructors and member methods, perform the following:

1. to accept and display the details of an account
2. to credit the account with some amount and display the message "A/C credited with Rs. XYZ and Balance: Rs. ABC" (where, XYZ is the amount credited and ABC is the new balance in the account).
3. to debit the account with some amount and display the message "A/C debited with Rs. XYZ and Balance: Rs. ABC" (where, XYZ is the amount debited and ABC is the new balance in the account).