LAB 3

Program to illustrate constructors

Definitions

Constructor – It is a method in a class which is called during the creation of the class object. The constructor is usually used to initialize values of properties. It can also be used to call methods or output some value during object creation. A constructor does not need to be called separately; it is called during object creation by the compiler. The name of the constructor must be the same as the class name, and the constructor has no return type. The constructor can not be called by us, it is called automatically only when the object is created.

Default constructor – The default constructor does not take any parameters. If we do not create our own constructor in the class, the default constructor is called automatically. We can overload the default constructor to initialize values.

Parameterized constructor – The parameterized constructor takes one or more parameters. These parameter values are used to assign our own values to the properties of an object. We can create multiple parameterized constructors using overloading, to suit our different needs.

Copy constructor – It is called when we try to create an object by taking values of an existing object. We need to create a copy constructor in our class before we use it.

Syntax : Class_name(Class_name instance_name) {

this.property_name = instance_name.property_name; }

Chaining of constructors – In constructor chaining, we call a constructor from within another constructor. We do this using the this keyword. We mostly use this concept in inheritance where we call a constructor of the superclass from the subclass. Which constructor is to be called using this keyword is determined by number and type of the parameters.

Code:

```java
/* Program to illustrate constructors */

import java.util.Scanner;

class Shape {
    double side, height;

    public Shape() {
        this(4, 0);

        System.out.println("Chaining back...");
        System.out.println("Inside the default constructor");
    }
```

```java
    public Shape(double S) {
        this(S, 0);
        System.out.println("Chaining back...");
        System.out.println("Inside the parameterized constructor");
        System.out.println("This constructor has 1 parameter");
    }

    public Shape(double S,  double H) {
        System.out.println("Inside the parameterized constructor");
        System.out.println("This constructor has 2 parameters");

        side = S;
        height = H;
    }

    Shape(Shape s) {
        System.out.println("Inside the copy constructor");

        this.side = s.side;
        this.height = s.height;
    }

    public void printArea() {
        if(this.height == 0) {
            System.out.println("The shape is a square");
            System.out.println("Area = " + side * side);
            System.out.println();
            return;
        }

        System.out.println("The shape is a cuboid");
        System.out.println("Volume = " + side * side * height);
        System.out.println();
    }
}

public class Lab3 {
    public static void main(String[] args) {

        try(Scanner sc = new Scanner(System.in)) {

            //Creating a shape using default constructor
            Shape S1 = new Shape();
            S1.printArea();

            //Creating a shape using parameterized constructor
            //Single parameter
            System.out.println("Enter value of side :");
            double side1 = sc.nextDouble();
            Shape S2 = new Shape(side1);
            S2.printArea();
```

```
            //Creating a shape using parameterized constructor
            //Two parameters
            System.out.println("Enter value of side and height:");
            double side2 = sc.nextDouble();
            double height2 = sc.nextDouble();
            Shape S3 = new Shape(side2, height2);
            S3.printArea();

            //Creating a shape using copy constructor
            //S4 is a copy of S3
            Shape S4 = new Shape(S3);
            S4.printArea();
        }
    }
}
```

Explanation:

- In this program, we have a class called shape. This class outputs the area of the shape if it is a square, or the volume if it is a cuboid. The cuboid will have height > 0 and square will have height = 0. By default, an object would be initialized as a square with side 4.
- The concept of constructor chaining has been used. The default constructor and the parameterized constructor with a single parameter call the constructor having 2 parameters to initialize the values of side and height. They both pass value of height as 0.
- First, we create an object S1 using the default constructor. The side is initialized to 4 and height to 0. Constructor with 2 parameters is called using constructor chaining. We can follow the function calls as below :

Shape S1 = new Shape(); -> call to the default constructor.

In the default constructor, we have the call this(4, 0); -> this will call constructor having 2 parameters.

The print statements are encountered and we get an output -> “Inside the parameterized constructor”

“This constructor has 2 parameters”

Then, values of side and height are initialized to 4 and 0 respectively.

Control is passed back to the default constructor, where we encounter more print statements that give us the output -> “Chaining back…”

“Inside the default constructor”

- We then call the printArea() method on object S1. First the method checks if it is a square or a cuboid. Since it is a square, the area is printed
- Next, we create an object S2 taking side input from the user. We initialise it using the parameterized constructor having one parameter. We have again used constructor chaining concept. printArea() method is called on S2 and since height is 0, it is a square. Area is printed.
- Object S3 is created as a cuboid by taking non zero height input from the user. If the user chooses to input height as 0, we will get a square again. The printArea function will print the volume of the cuboid S3.
- Object S4 is crested using the copy constructor. It is copied from object S3.

Output:

```
PS D:\NITJ\Sem 4\Java\Lab 3> javac Lab3.java
PS D:\NITJ\Sem 4\Java\Lab 3> java Lab3
Inside the parameterized constructor
This constructor has 2 parameters
Chaining back...
Inside the default constructor
The shape is a square
Area = 16.0

Enter value of side :
6
Inside the parameterized constructor
This constructor has 2 parameters
Chaining back...
Inside the parameterized constructor
This constructor has 1 parameter
The shape is a square
Area = 36.0

Enter value of side and height:
3 7
Inside the parameterized constructor
This constructor has 2 parameters
The shape is a cuboid
Volume = 63.0

Inside the copy constructor
The shape is a cuboid
Volume = 63.0

PS D:\NITJ\Sem 4\Java\Lab 3>
```