Nithin S
221IT085

IT150 Lab Assignment 9

Q1.Write a Java application that prompts the user for side length of a square and uses a method
called SquareArea to calculate the area of the square and uses a method SquarePerimeter to
calculate the perimeter of the square.
a) In the above program declare the side length with ‘a’ and create multiple objects
to create the area and perimeter of the square.
b) Create a Rectangle class equivalent to the Square class created as above. Before
coding the class, decide which variables and methods this class must have.
c) Create a class called TestQuadrilaters that prints the details of a square and a rectangle

Code

TestQuadrilaterals.java > ...

```java
import java.util.Scanner;

class Square {
    private double sideLength;

    public Square(double sideLength) {
        this.sideLength = sideLength;
    }

    public double calculateArea() {
        return sideLength * sideLength;
    }

    public double calculatePerimeter() {
        return 4 * sideLength;
    }
}

class Rectangle {
    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    public double calculateArea() {
        return length * width;
    }

    public double calculatePerimeter() {
        return 2 * (length + width);
    }
}

public class TestQuadrilaterals {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print(s:"Enter side length of square: ");
        double sideLength = scanner.nextDouble();
        Square square = new Square(sideLength);

        System.out.print(s:"Enter length of rectangle: ");
        double length = scanner.nextDouble();
        System.out.print(s:"Enter width of rectangle: ");
        double width = scanner.nextDouble();
```

```java
public class TestQuadrilaterals {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print(s:"Enter side length of square: ");
        double sideLength = scanner.nextDouble();
        Square square = new Square(sideLength);

        System.out.print(s:"Enter length of rectangle: ");
        double length = scanner.nextDouble();
        System.out.print(s:"Enter width of rectangle: ");
        double width = scanner.nextDouble();
        Rectangle rectangle = new Rectangle(length, width);

        System.out.println("Square Area: " + square.calculateArea());
        System.out.println("Square Perimeter: " + square.calculatePerimeter());

        System.out.println("Rectangle Area: " + rectangle.calculateArea());
        System.out.println("Rectangle Perimeter: " + rectangle.calculatePerimeter());

        scanner.close();
    }
}
```

OUTPUT

```
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_10$ javac TestQuadrilaterals.java
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_10$ java TestQuadrilaterals
Enter side length of square: 7
Enter length of rectangle: 5
Enter width of rectangle: 4
Square Area: 49.0
Square Perimeter: 28.0
Rectangle Area: 20.0
Rectangle Perimeter: 18.0
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_10$
```

Q2.Write a JAVA program to create a Mydetails with data members Roll no, name and department.
The constructor Mydetails () should initialize the Roll no, name and the department. The member
function display() should display the values of Roll no, name and the department. In the main()
method, an object ‘m’ of class Mydetails to be created with values 1010,”xxx” and “xxdeptxx”.
Then the display() method should be called.

Code

```
class Mydetails {
    private int rollNo;
    private String name;
    private String department;

    public Mydetails(int rollNo, String name, String department) {
        this.rollNo = rollNo;
        this.name = name;
        this.department = department;
    }

    public void display() {
        System.out.println("Roll No: " + rollNo);
        System.out.println("Name: " + name);
        System.out.println("Department: " + department);
    }
}

public class Main {
    Run | Debug
    public static void main(String[] args) {
        Mydetails m = new Mydetails(rollNo:1010, name:"xxx", department:"xxdeptxx");
        m.display();
    }
}
```

OUTPUT

```
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_10$ javac Main.java
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_10$ java M
Main       Mydetails
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_10$ java Main
Roll No: 1010
Name: xxx
Department: xxdeptxx
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_10$
```

Q3.Write a JAVA program using this keyword to refer the current class instance variable.

CODE

This.java > ...

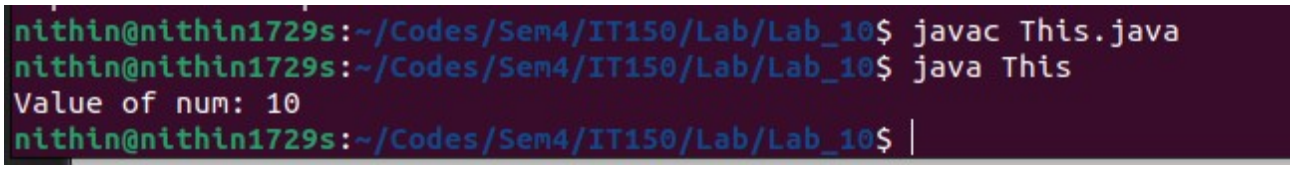
```java
class Example {
    private int num;

    public Example(int num) {
        this.num = num; // 'this' refers to the current instance of t
    }

    public void display() {
        System.out.println("Value of num: " + this.num);
    }
}

public class This {
    Run | Debug
    public static void main(String[] args) {
        Example obj = new Example(num:10);
        obj.display();
    }
}
```

OUTPUT



Q4.Write a JAVA program with Multiple constructors (constructor overloading) where
i. 1
st constructor is with multiple parameters,
ii. 2
nd constructor with one parameter defined and others as default values.
iii. 3
rd constructor in terms of another object.
iv. Default constructor (without any arguments)
☔ In the above program, use this keyword for multiple constructors and obtain the output.
Provide your explanation of the outcomes with and without the use this keyword

CODE

Mul.java > Mul

```java
class MultipleConstructors {
    private int a;
    private int b;

    // Constructor with multiple parameters
    public MultipleConstructors(int a, int b) {
        this.a = a;
        this.b = b;
    }

    // Constructor with one parameter and others default
    public MultipleConstructors(int a) {
        this(a, b:0); // Using 'this' to call another constructor
    }

    // Constructor in terms of another object
    public MultipleConstructors(MultipleConstructors obj) {
        this.a = obj.a;
        this.b = obj.b;
    }

    // Default constructor
    public MultipleConstructors() {
        this(a:0, b:0); // Using 'this' to call another constructor
    }

    public void display() {
        System.out.println("Value of a: " + this.a);
        System.out.println("Value of b: " + this.b);
    }
}

public class Mul {
    Run | Debug
    public static void main(String[] args) {
        MultipleConstructors obj1 = new MultipleConstructors(a:10, b:20);
        obj1.display();

        MultipleConstructors obj2 = new MultipleConstructors(a:30);
        obj2.display();

        MultipleConstructors obj3 = new MultipleConstructors(obj1);
        obj3.display();

        MultipleConstructors obj4 = new MultipleConstructors();
        obj4.display();
    }
}
```

OUTPUT


nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_10$ javac Mul.java
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_10$ java Mul
Value of a: 10
Value of b: 20
Value of a: 30
Value of b: 0
Value of a: 10
Value of b: 20
Value of a: 0
Value of b: 0
nithin@nithin1729s:~/Codes/Sem4/IT150/Lab/Lab_10$