Name: Chandrahasa B

Student code: AF0336567

Batch Code: ANP-C6315

Lab Assignment – 16

Interface 2

Question 1: Create an abstract class Shape with an abstract method calculateArea(). Implement two subclasses, Circle and Rectangle, which inherit from Shape and provide their own implementations of calculateArea().

Write a program to calculate and print the areas of a circle and a rectangle.

Input:

```
// Abstract Shape class
abstract class Shape {
    // Abstract method to calculate area
    abstract double calculateArea();
}

// Circle subclass
class Circle extends Shape {
    double radius;

    // Constructor
    Circle(double radius) {
        this.radius = radius;
    }

    // Implementation of calculateArea for Circle
    @Override
    double calculateArea() {

        return Math.PI * radius * radius;

    }

}


// Rectangle subclass

class Rectangle extends Shape {

    double length;

    double width;


    // Constructor

    Rectangle(double length, double width) {

        this.length = length;

        this.width = width;

    }


    // Implementation of calculateArea for Rectangle

    @Override

    double calculateArea() {

        return length * width;

    }

}


// Main program

public class Main {

    public static void main(String[] args) {

        // Creating a Circle with radius 5
        Circle circle = new Circle(5);

        // Creating a Rectangle with length 4 and width 6
        Rectangle rectangle = new Rectangle(4, 6);

        // Calculating and printing the areas
        System.out.println("Area of Circle: " + circle.calculateArea());
        System.out.println("Area of Rectangle: " + rectangle.calculateArea());
    }
}
```

Output:

```
Area of Circle: 78.53981633974483

Area of Rectangle: 24.0
```

Question 2: Write a Java program that demonstrates method overriding by creating a superclass called Animal and two subclasses called Dog and Cat.

● The Animal class should have a method called makeSound(), which simply prints "The animal makes a sound."

● The Dog and Cat classes should override this method to print "TheCat/The dog meows/barks" respectively.

● The program should allow the user to create and display objects of each class. [Hint:Use multilevel inheritance]

Input:

```
// Superclass
class Animal {
    // Method in the superclass
    void makeSound() {
        System.out.println("The animal makes a sound.");
    }
```

```
}

// Subclass Dog
class Dog extends Animal {
    // Override the makeSound method for Dog
    @Override
    void makeSound() {
        System.out.println("The dog barks.");
    }
}

// Subclass Cat
class Cat extends Animal {
    // Override the makeSound method for Cat
    @Override
    void makeSound() {
        System.out.println("The cat meows.");
    }
}

// Main program
public class Main {
    public static void main(String[] args) {
        // Create objects of each class
        Animal genericAnimal = new Animal();
        Dog myDog = new Dog();
        Cat myCat = new Cat();
        // Demonstrate method overriding
        System.out.println("Generic Animal:");
        genericAnimal.makeSound();

        System.out.println("\nMy Dog:");
        myDog.makeSound();

        System.out.println("\nMy Cat:");
        myCat.makeSound();
    }
}
```

Output:

```
Generic Animal:
The animal makes a sound.
y Dog:
The dog barks.
My Cat:
The cat meows.
```