**Session-2 Lab**

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.

**CODE:**

package Abstractinterface;

abstract class Shape {

abstract double calculateArea();

}

class Circle extends Shape {

double radius;

Circle(double radius) {

this.radius = radius;

}

*@Override*

double calculateArea() {

return Math.*PI* \* radius \* radius;

}

}

class Rectangle extends Shape {

double length, width;

Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

*@Override*

double calculateArea() {

return length \* width;

}

}

public class ShapeDemo {

public static void main(String[] args) {

Shape circle = new Circle(35.5);

Shape rectangle = new Rectangle(7.0, 6.0);

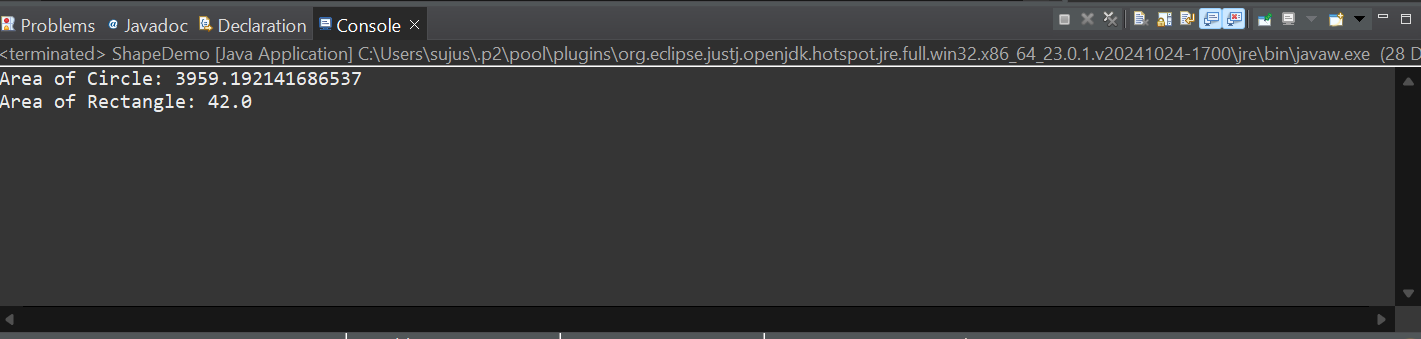
System.*out*.println("Area of Circle: " + circle.calculateArea());

System.*out*.println("Area of Rectangle: " + rectangle.calculateArea());

}

}

**OUTPUT**

****

1. Write a Java program that demonstrates method overriding by creating a superclass called Animal and two subclasses called Dog and Cat.
   1. The Animal class should have a method called makeSound(), which simply prints "The animal makes a sound."
   2. The Dog and Cat classes should override this method to print "The dog barks" and "The cat meows," respectively.
   3. The program should allow the user to create and display objects of each class.

*Hint: Use multilevel inheritance.*

**CODE**

package Abstractinterface;

class Animal {

void makeSound() {

System.*out*.println("The animal makes a sound.");

}

}

class Dog extends Animal {

*@Override*

void makeSound() {

System.*out*.println("The dog barks.");

}

}

class Cat extends Animal {

*@Override*

void makeSound() {

System.*out*.println("The cat meows.");

}

}

public class AnimalDemo {

public static void main(String[] args) {

Animal animal = new Animal();

Animal dog = new Dog();

Animal cat = new Cat();

animal.makeSound();

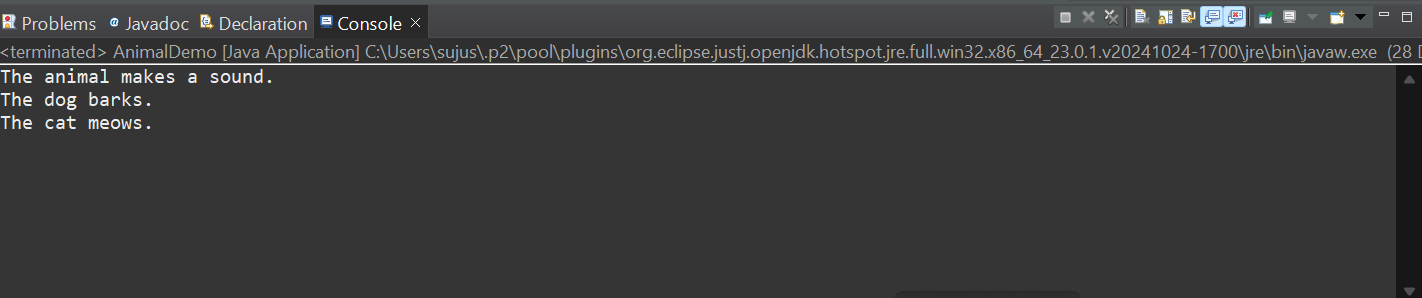
dog.makeSound();

cat.makeSound();

}

}

**OUTPUT**

****