**1. Create a superclass called Animal with a method makeSound() that prints the sound made by the animal. Implement subclasses Dog, Cat, and Cow that inherit from the Animal class. Implement the makeSound() method in each subclass to print the sound made by a dog, cat, and cow, respectively.**

// Superclass Animal

class Animal {

public void makeSound() {

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

}

}

// Subclass Dog

class Dog extends Animal {

@Override

public void makeSound() {

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

}

}

// Subclass Cat

class Cat extends Animal {

public void makeSound() {

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

}

}

// Subclass Cow

class Cow extends Animal {

public void makeSound() {

System.out.println("The cow moos");

}

}

// Example usage

public class Main {

public static void main(String[] args) {

Animal animal = new Animal();

animal.makeSound();

Dog dog = new Dog();

dog.makeSound();

Cat cat = new Cat();

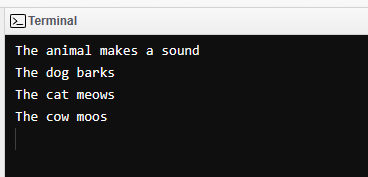
cat.makeSound();

Cow cow = new Cow();

cow.makeSound();

}

}



**2. Create a superclass called Shape with an abstract method calculateArea() that returns the area of the shape. Implement subclasses Rectangle, Circle, and Triangle that inherit from the Shape class. Implement the calculateArea() method in each subclass to calculate and return the area of a rectangle, circle, and triangle, respectively. Then, create a class called ShapeCalculator with a method printArea(Shape shape) that accepts an object of type Shape and prints its area. Demonstrate polymorphism by passing instances of different subclasses to the printArea() method.**

**// Superclass Shape**

**abstract class Shape {**

**public abstract double calculateArea();**

**}**

**// Subclass Rectangle**

**class Rectangle extends Shape {**

**private double length;**

**private double width;**

**public Rectangle(double length, double width) {**

**this.length = length;**

**this.width = width;**

**}**

**@Override**

**public double calculateArea() {**

**return length \* width;**

**}**

**}**

**// Subclass Circle**

**class Circle extends Shape {**

**private double radius;**

**public Circle(double radius) {**

**this.radius = radius;**

**}**

**@Override**

**public double calculateArea() {**

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

**}**

**}**

**// Subclass Triangle**

**class Triangle extends Shape {**

**private double base;**

**private double height;**

**public Triangle(double base, double height) {**

**this.base = base;**

**this.height = height;**

**}**

**@Override**

**public double calculateArea() {**

**return 0.5 \* base \* height;**

**}**

**}**

**// ShapeCalculator class**

**class ShapeCalculator {**

**public void printArea(Shape shape) {**

**double area = shape.calculateArea();**

**System.out.println("The area of the shape is: " + area);**

**}**

**}**

**// Example usage**

**public class Main {**

**public static void main(String[] args) {**

**ShapeCalculator calculator = new ShapeCalculator();**

**Rectangle rectangle = new Rectangle(5.0, 3.0);**

**calculator.printArea(rectangle);**

**Circle circle = new Circle(4.0);**

**calculator.printArea(circle);**

**Triangle triangle = new Triangle(6.0, 2.5);**

**calculator.printArea(triangle);**

**}**

**}**

**Output:**

**The area of the shape is: 15.0**

**The area of the shape is: 50.26548245743669**

**The area of the shape is: 7.5**

**3. Create a class called Person with private properties like name, age, and address. Provide public getter and setter methods for these properties. Write a program that creates an instance of the Person class, sets values for its properties using the setter methods, and displays the values using the getter methods.**

**class Person {**

**private String name;**

**private int age;**

**private String address;**

**// Getter and Setter for Name**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**// Getter and Setter for Age**

**public int getAge() {**

**return age;**

**}**

**public void setAge(int age) {**

**this.age = age;**

**}**

**// Getter and Setter for Address**

**public String getAddress() {**

**return address;**

**}**

**public void setAddress(String address) {**

**this.address = address;**

**}**

**}**

**public class Main {**

**public static void main(String[] args) {**

**// Creating an instance of Person**

**Person person = new Person();**

**// Setting values using setter methods**

**person.setName("John Doe");**

**person.setAge(25);**

**person.setAddress("123 Main St");**

**// Getting values using getter methods**

**String name = person.getName();**

**int age = person.getAge();**

**String address = person.getAddress();**

**// Displaying the values**

**System.out.println("Name: " + name);**

**System.out.println("Age: " + age);**

**System.out.println("Address: " + address);**

**}**

**}**

**Output:**

**Name: John Doe**

**Age: 25**

**Address: 123 Main St**

**4. Create an interface called Drawable with a method draw() that has no implementation. Implement this interface in classes Circle and Rectangle. Write a program that creates objects of Circle and Rectangle and calls the draw() method on each object.**

**// Drawable interface**

**interface Drawable {**

**void draw();**

**}**

**// Circle class implementing the Drawable interface**

**class Circle implements Drawable {**

**@Override**

**public void draw() {**

**System.out.println("Drawing a circle");**

**}**

**}**

**// Rectangle class implementing the Drawable interface**

**class Rectangle implements Drawable {**

**@Override**

**public void draw() {**

**System.out.println("Drawing a rectangle");**

**}**

**}**

**// Example usage**

**public class Main {**

**public static void main(String[] args) {**

**// Creating objects of Circle and Rectangle**

**Circle circle = new Circle();**

**Rectangle rectangle = new Rectangle();**

**// Calling the draw() method on each object**

**circle.draw();**

**rectangle.draw();**

**}**

**}**

**Output:**

**Drawing a circle**

**Drawing a rectangle**