Abstract classes and Interfaces

Assignment-1

```java
package Abstract;



// Abstract class Shape
abstract class Shape {

// Abstract method to calculate the area
public abstract double calculateArea();
}

// Circle class inheriting from Shape
class Circle extends Shape {

private double radius;

// Constructor to initialize the radius of the circle
public Circle(double radius) {

this.radius = radius;
}

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

return Math.PI * radius * radius; // Area of a circle: πr²
}
}

// Rectangle class inheriting from Shape
class Rectangle extends Shape {

private double width;
private double height;

// Constructor to initialize the width and height of the rectangle
public Rectangle(double width, double height) {
```

```
        this.width = width;
        this.height = height;
    }

    // Implementation of calculateArea for a Rectangle
    @Override
    public double calculateArea() {
        return width * height; // Area of a rectangle: width * height
    }
}

public class Rectangle1 {

    // Main class to test the functionality
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        // Create a Circle object with radius 5
        Shape circle = new Circle(5);
        System.out.println("Area of Circle: " + circle.calculateArea());

        // Create a Rectangle object with width 4 and height 7
        Shape rectangle = new Rectangle(4, 7);
        System.out.println("Area of Rectangle: " + rectangle.calculateArea());

    }

}
```

Output

```
Area of Circle: 78.53981633974483
Area of Rectangle: 28.0
```

Assignment-2

```
package Abstract;
```

```java
//Superclass Animal
class Animal {

// Method that will be overridden in subclasses
public void makeSound() {

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

//Subclass Dog inheriting from Animal
class Dog extends Animal {

// Overriding makeSound method for Dog
@Override
public void makeSound() {

System.out.println("The Dog barks");
}
}

//Subclass Cat inheriting from Animal
class Cat extends Animal {

// Overriding makeSound method for Cat
@Override
public void makeSound() {

System.out.println("The Cat meows");
}
}

public class MultilevelInheritance {

public static void main(String[] args) {
// TODO Auto-generated method stub

// Create a Dog object
Animal myDog = new Dog();
myDog.makeSound(); // The Dog barks

// Create a Cat object
Animal myCat = new Cat();
myCat.makeSound(); // The Cat meows
```

```
        // Create an Animal object
        Animal myAnimal = new Animal();
myAnimal.makeSound(); // The animal makes a sound

    }

    }
```

Output

```
The Dog barks
The Cat meows
The animal makes a sound
```