Name: S.M.B.Dissanayake

Id:27357

Exercise 1:

Class 01: The first class is named Student, which is declared as final. When a class is declared as final, it means it cannot be subclassed (i.e., cannot have any subclasses). So, the class Student cannot be extended by any other class. The class has a final integer variable mark, which means it's a constant and cannot be changed once initialized. It is initialized with the value 100. The class also has a final method display(), which means it cannot be overridden by subclasses. The method prints the value of the marks variable.

Class 02: The second class is named Undergraduate, which attempts to extend the Student class. However, this will result in a compilation error. Since the Student class is declared as final, it cannot be subclassed, and any attempt to do so will lead to a compilation error. So, the outcome of trying to compile these classes will result in an error due to the attempt to extend a final class in Class 02.

To fix this, you can remove the final keyword from the Student class if you intend to create subclasses. However, if you want to maintain the immutability of the display() method or the marks variable, you may keep them as final as per your requirements.

Exercise 2:

```
// Abstract Class: Shape
abstract class Shape {
    // Abstract method: calculateArea
    public abstract double calculateArea();

    // Non-abstract method: display
    public void display() {
        System.out.println("This is a shape.");
    }
```

```
}

// Subclass: Circle
class Circle extends Shape {
    private double radius;

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

    // Implementation of calculateArea for Circle
    @Override
    public double calculateArea() {
        return Math.PI * radius * radius;
    }
}

// Subclass: Rectangle
class Rectangle extends Shape {
    private double length;
    private double width;

    // Constructor for Rectangle
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    // Implementation of calculateArea for Rectangle
    @Override
    public double calculateArea() {
        return length * width;
    }
}

// Main class
public class Main {
    public static void main(String[] args) {
        // Instantiating Circle with a radius of 5 units
        Circle circle = new Circle(5);
        System.out.println("Area of the circle: " + circle.calculateArea());
        circle.display();

        // Instantiating Rectangle with length 6 units and width 4 units
        Rectangle rectangle = new Rectangle(6, 4);
        System.out.println("Area of the rectangle: " + rectangle.calculateArea());
        rectangle.display();
    }
}
```