Exercise 01:

Try following code. What is the outcome? Why?

Class 01: Class 02:

final class Student { class Undergraduate extends Student{}

final int marks = 100;

final void display();

}

*The code you provided will result in a compilation error.*

*In class 01,* ***Student*** *is declared as a* ***final*** *class, which means it cannot be extended by any other class. It also has a* ***final*** *instance variable* ***marks*** *initialized with a value of 100. This variable cannot be reassigned once it's initialized due to the* ***final*** *keyword.*

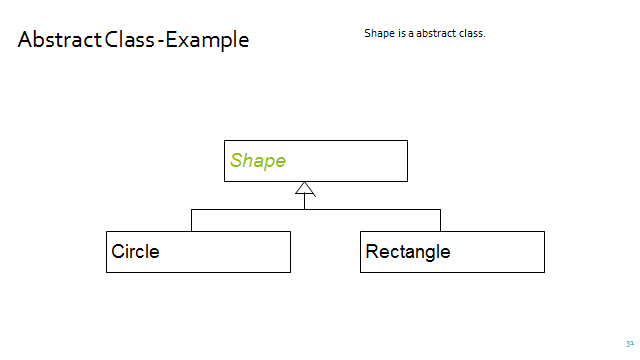
*However, the issue lies with the* ***final*** *method* ***display()****. The* ***final*** *modifier on a method means it cannot be overridden by any subclass. But in this case, the method* ***display()*** *is declared without any implementation (missing method body). Since it is* ***final*** *and does not have an implementation, it will cause a compilation error.*

*In class 02,* ***Undergraduate*** *attempts to extend the* ***Student*** *class. However, since the* ***Student*** *class is declared as* ***final****, it cannot be subclassed. Therefore, extending the* ***Student*** *class by creating the* ***Undergraduate*** *class will also result in a compilation error.*

*To fix these errors, you can either remove the* ***final*** *keyword from the* ***Student*** *class declaration, provide an implementation for the* ***display()*** *method, or remove the* ***extends Student*** *part from the* ***Undergraduate*** *class if the intention is not to subclass* ***Student****.*

Exercise 02:

Develop a code base for the following scenario. Shape class contains an abstract method called “calculateArea” and non-abstract method called “display”. Try to pass required values at the instantiation. Recall what we have done at the lecture…



public abstract class Shape {

abstract double calculateArea();  
  
 public void display(){  
 System.*out*.println("Area: "+calculateArea());  
 }  
}

public class Circle extends Shape{

private int radius;  
  
 public Circle(int radius){  
 this.radius = radius;  
 }  
 @Override  
 double calculateArea(){  
 return Math.*PI*\*radius\*radius;  
 }  
}

public class Rectangle extends Shape{

private int width;  
 private int height;  
  
 public Rectangle(int width, int height){  
 this.height = height;  
 this.width = width;  
 }  
 @Override  
 double calculateArea() {  
 return height\*width;  
 }  
}

public class FindArea {

public static void main(String[] args){  
 var circle1 = new Circle(7);  
 circle1.display();  
  
 var rectangle1 = new Rectangle(5,6);  
 rectangle1.display();  
  
 }

}