Exercise 01:

Try following code. What is the outcome? Why?

Class 01:

```
final class Student {

        final int marks = 100;

        final void display();

}
```

Class 02:

```
class Undergraduate extends Student{}
```

Answers

The outcome of in this code compilation error. The reason for this is that the Student class is declared as final, which means that it cannot be inherited from. However, the Undergraduate class inherits from the Student class, which is not allowed.

```
final class Student {

   final int marks = 100;

   final void display() {

      System.out.println("Marks: " + marks);

   }

}

class Undergraduate extends Student() {

}
```

Correct code

Why?

The final keyword prevents the Student class from being inherited from. This is because the final keyword is used to indicate that a class cannot be changed. If the Student class were not declared as final, then the Undergraduate class would be able to inherit from it.

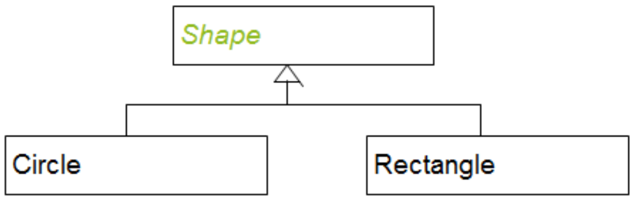
However, the Student class is declared as final, so the Undergraduate class cannot inherit from it. This is why the code will not compile.

<u>Exercise 02:</u>

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…

Abstract Class-Example

Shape is a abstract class.



<u>Answers</u>

```
abstract class Shape {
    abstract double calculateArea();


    void display() {
        System.out.println("This is a shape.");
    }
}


class Circle extends Shape {
    private double radius;
    Circle(double radius) {
        this.radius = radius;
    }
    @Override
    double calculateArea() {
        return Math.PI * radius * radius;
    }
}


class Rectangle extends Shape {


    private double width;
    private double height;


    Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }


    @Override
    double calculateArea() {
        return width * height;
    }
}


public class Main {
    public static void main(String[] args) {
        Circle circle = new Circle(5);
        Rectangle rectangle = new Rectangle(10, 20);


        circle.display();
        System.out.println("The area of the circle is: " + circle.calculateArea());


        rectangle.display();
        System.out.println("The area of the rectangle is: " +
rectangle.calculateArea());
    }
}
```