Practical 07: Inheritance & Abstract Classes

Exercise 01:

Try following code. What is the outcome? Why?

Class 01:

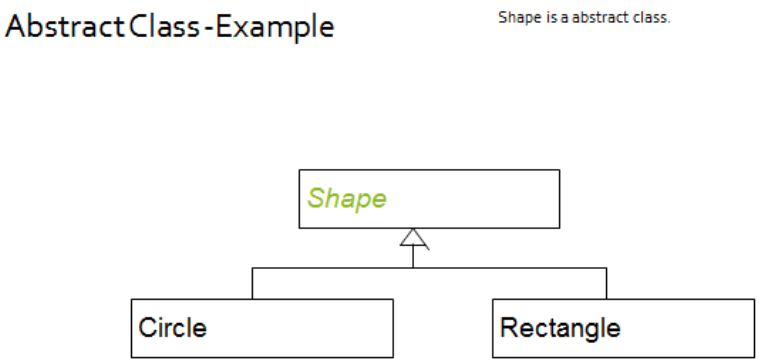
```
final class Student {
        final int marks = 100;
        final void display();
}
```

Class 02:

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

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…




```
abstract class Shape {
   abstract double calculateArea();
   void display() {
      System.out.println("Area: " + calculateArea());
   }
}
```

```
class Circle extends Shape {
    private double radius;

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

    @Override
    double calculateArea() {
        return Math.PI * radius * radius;
    }
}

class Rectangle extends Shape {
    private double length;
    private double width;

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

    @Override
```

```
    double calculateArea() {
        return length * width;
    }
}

class MainClass {
    public static void main(String[] args) {
        Circle circle = new Circle(5);
        circle.display();

        Rectangle rectangle = new Rectangle(10, 5);
        rectangle.display();
    }
}
```