

1.

Try following code. What is the outcome? Why?

Class 01:

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

Class 02:

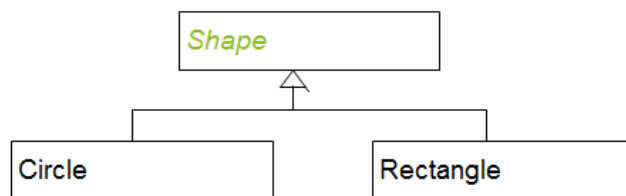
```
class Undergraduate extends Student{}
```

2.

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...

AbstractClass-Example

Shape is a abstract class.



```
package com.mycompany.Labs;
```

```
public class TestShape
```

```
{
```

```
    public static void main(String[] args)
```

```
{
```

```
    double circleRadius = 5.0;
```

```
    Circle circle = new Circle(circleRadius);
```

```
        circle.display();
    }
}
```

```
package com.mycompany.Labs;

abstract class Shape
{
    public abstract double calculateArea();

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

```
package com.mycompany.Labs;

class Circle extends Shape
{
    private double radius;

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

    @Override
```

```
public double calculateArea() {  
    return Math.PI * radius * radius;  
}  
}
```