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();

}

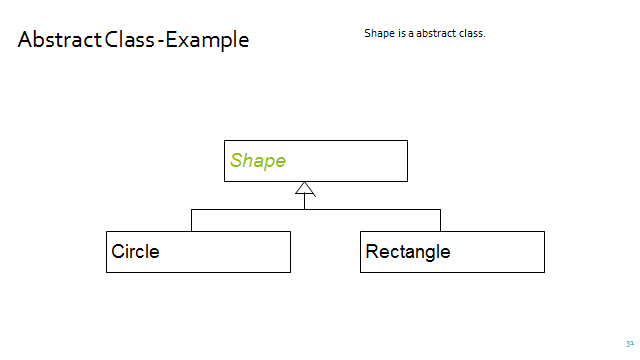
out come : compilation error

why:

* if we declares the “student” class as “final’,it cann’t be create any sub classes.
* if we declares the final variable like above integer variable “marks”,it cann’t be ressigned a new value once it is initialized.
* “final void display();” ,this is a abtract method declaration. however, abstract method cann’t be marked as “final”, if we marked like this, declaration result in a compilation error.

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…



package com.mycompany.java05;

public abstract class Shape {

public abstract void calculateArea();

public void display()

{

}

}

package com.mycompany.java05;

public class Rectangal extends Shape {

private int width;

private int length;

public Rectangal(int width,int length)

{

this.width=width;

this.length=length;

}

public void calculateArea()

{

System.out.println("area: "+(width\*length));

}

}

package com.mycompany.java05;

public class Circle extends Shape {

private double radius;

public Circle(double radius)

{

this.radius=radius;

}

public void calculateArea()

{

System.out.println("area: "+(Math.PI\*radius));

}

}

package com.mycompany.java05;

public class Java05 {

public static void main(String[] args) {

Rectangal r=new Rectangal(12,5);

Circle c=new Circle(14);

r.calculateArea();

c.calculateArea();

}

}