

Lab Exercise 2

Programs to implement different types of Inheritance. Implement the following:

- a. An abstract class "Shape" with the following properties: an instance variable shapeName of type String, an abstract method area (), a toString() method that returns the name of the shape.
- b. Create a subclass named "Sphere" which has radius and its area given by the formula $4 \cdot \text{PI} \cdot r^2$.
- c. Create a subclass named "Rectangle" which has length and width and its area is length times width.
- d. Create a subclass named "Triangle" which has base and height and its area is $\frac{1}{2} \cdot \text{base} \cdot \text{height}$.
- e. Create another class which displays the calculated area.

Theory: An abstract class is a class that is declared abstract. It may or may not include abstract methods. Abstract classes cannot be instantiated, but they can be subclassed. An abstract method is a method that is declared without an implementation (without braces, and followed by a semicolon), like this: `abstract void moveTo(double deltaX, double deltaY);`

If a class includes abstract methods, then the class itself *must* be declared abstract. Objects of an abstract class cannot be created. Even then, constructors may be provided for an abstract class for use by its sub classes. An abstract class is made usable by extending it to a sub class. The sub class needs to provide implementations of all the abstract methods defined in its super class, just in the way methods are overridden.

```
import java.util.Scanner;

abstract class Shape
{
    final double PI=3.14; //constant
    String shapeName;
    abstract void area(); //abstract method

    public String toString() //toString method to return calling classname
    {
        return this.getClass().getName();
    }
}
```

```
class Sphere extends Shape
{
    double rad;
    Sphere(double r) //constructor to set radius
    {
        rad=r;
    }
    void area()
    {
        System.out.println("Area = "+(4*PI*rad*rad)); //area of sphere formula
    }
}
```

```
class Rectangle extends Shape
{
    double len, width;

    Rectangle(double l,double w)
    {
        len=l;
        width=w;
    }
    void area()
    {
        System.out.println("Area = "+(len*width));
    }
}
```

```
class Triangle extends Shape
{
    double    base, height;

    Triangle(double b,double h)
    {
        base=b;
```

```

        height=h;
    }
    void area()
    {
        System.out.println("Area = "+(0.5*base*height));
    }
}

```

```

public class Three {
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the radius of a circle");
        double r=s.nextInt(); //taking radius input
        Sphere sp=new Sphere(r); //sphere object creation
        System.out.println(sp.toString()); //respective class object calls toString()
        sp.area(); //calling respective class area method
        System.out.println("Enter the len and wid: ");
        double l=s.nextInt();
        double w=s.nextInt();
        Rectangle rc=new Rectangle(l,w);
        System.out.println(rc.toString());
        rc.area();
        System.out.println("Enter the base and height: ");
        double b=s.nextInt();
        double h=s.nextInt();
        Triangle ta=new Triangle(b,h);
        System.out.println(ta.toString());
        ta.area();
    }
}

```

