

Painting Shapes

In this lab exercise you will develop a class hierarchy of shapes and write a program that computes the amount of paint needed to paint different objects. The hierarchy will consist of a parent class Shape with three derived classes - Sphere, Rectangle, and Cylinder. For the purposes of this exercise, the only attribute a shape will have is a name and the method of interest will be one that computes the area of the shape (surface area in the case of three-dimensional shapes). Do the following.

1. Write 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
2. The file *Sphere.java* contains a class for a sphere which is a descendant of Shape. A sphere has a radius and its area (surface area) is given by the formula $4\pi r^2$. Define similar classes for a rectangle and a cylinder. Both the Rectangle class and the Cylinder class are descendants of the Shape class. A rectangle is defined by its length and width and its area is length times width. A cylinder is defined by a radius and height and its area (surface area) is $2\pi r^2 + 2\pi rh$. Define the `toString` method in a way similar to that for the Sphere class.
3. The file *Paint.java* contains a class for a type of paint (which has a "coverage" and a method to compute the amount of paint needed to paint a shape). Correct the return statement in the amount method so the correct amount will be returned. Use the fact that the amount of paint needed is the area of the shape divided by the coverage for the paint. (NOTE: Leave the print statement - it is there for illustration purposes, so you can see the method operating on different types of Shape objects.)
4. The file *PaintThings.java* contains a program that computes the amount of paint needed to paint various shapes. A paint object has been instantiated. Add the following to complete the program:
 - ☐ Instantiate the three shape objects: deck to be a 20 by 35 foot rectangle, bigBall to be a sphere of radius 15, and tank to be a cylinder of radius 10 and height 30.
 - ☐ Make the appropriate method calls to assign the correct values to the three amount variables.
 - ☐ Run the program and test it. You should see polymorphism in action as the amount method computes the amount of paint for various shapes.

```
//*****
// Sphere.java
//
// Represents a sphere.
//*****
public class Sphere extends Shape
{
    private double radius; //radius in feet

    //-----
    // Constructor: Sets up the sphere.
    //-----
    public Sphere(double r)
    {
        super("Sphere");
        radius = r;
    }

    //-----
    // Returns the surface area of the sphere.
    //-----
    public double area()
    {
        return 4*Math.PI*radius*radius;
    }
}
```

```

//-----
// Returns the sphere as a String.
//-----
public String toString()
{
    return super.toString() + " of radius " + radius;
}

}

//*****
// Paint.java
//
// Represents a type of paint that has a fixed area
// covered by a gallon. All measurements are in feet.
//*****

public class Paint
{
    private double coverage; //number of square feet per gallon

    //-----
    // Constructor: Sets up the paint object.
    //-----
    public Paint(double c)
    {
        coverage = c;
    }

    //-----
    // Returns the amount of paint (number of gallons)
    // needed to paint the shape given as the parameter.
    //-----
    public double amount(Shape s)
    {
        System.out.println ("Computing amount for " + s);
        return 0;
    }
}

```

```

//*****
//  PaintThings.java
//
//  Computes the amount of paint needed to paint various
//  things. Uses the amount method of the paint class which
//  takes any Shape as a parameter.
//*****

import java.text.DecimalFormat;

public class PaintThings
{
    //-----
    // Creates some shapes and a Paint object
    // and prints the amount of paint needed
    // to paint each shape.
    //-----
    public static void main (String[] args)
    {
        final double COVERAGE = 350;
        Paint paint = new Paint(COVERAGE);

        Rectangle deck;
        Sphere bigBall;
        Cylinder tank;

        double deckAmt, ballAmt, tankAmt;

        // Instantiate the three shapes to paint

        // Compute the amount of paint needed for each shape

        // Print the amount of paint for each.
        DecimalFormat fmt = new DecimalFormat("0.##");
        System.out.println ("\nNumber of gallons of paint needed...");
        System.out.println ("Deck " + fmt.format(deckAmt));
        System.out.println ("Big Ball " + fmt.format(ballAmt));
        System.out.println ("Tank " + fmt.format(tankAmt));
    }
}

```