Week 3: Outline

Reading:
1. Dale, Chapter 7
2. Dale, Lab 3

Objectives:
1. Resolve scope discussion
2. Discuss Access
3. Introduce abstract classes

Concepts:
1. Scope of Access
2. Use of Access
3. Abstract classes basics
1. Resolve scope discussion
   - Finish scope discussion
   - Show scoping examples
   - Answer additional questions

2. Discuss Access
   - Basics of Access use

3. Introduce abstract classes
   - Abstract classes basics
   - Abstract superclasses
   - Concrete classes
   - Difference between abstract and concrete classes

4. Lab 3 Overview
Objective:
1. Practice inheritance

Student Activities:
1. Inheritance: Point, Circle, Cylinder
   We consider a point, circle, cylinder hierarchy. First we develop and use class Point. The we present an example in which we derive class Circle from class Point. Finally, we present an example in which we derive class Cylinder from class Circle. You must create the Point.java, Circle.java, and Cylinder.java. The test cases for all these are given and so are the results.

   a) Create Point.java. Create public point class. Point’s instance variables are protected. Thus, when class Circle is derived from class Point, the methods of class Circle will be able to reference coordinates x and y directly rather than using access methods. Class Point include methods setPoint, getX, getY, toString and two Point constructors. One constructor (no argument constructor) must call class Object’s constructor. Inside setPoint(0.0). The other constructor calls in parameters xCoordinate and yCoordinate and puts them into the setPoint(xCoordinate, yCoordinate). One method is setPoint and it passes in the xCoordinate and yCoordinate parameters and sets them to x and y. The other two are simply get methods getx() and gety(). The last is a String toString() and it returns \\
   "[“ + x + “, “ + y + “]”;

   GIVEN: TESTP.JAVA // To test point.java
   Test.java
   // Applet to test class Point
   // Java extension packages
   import javax.swing.JOptionPane;
   // Deitel packages
   import com.deitel.jhtp4.ch09.Point;
   public class Test {
   // test class Point
   public static void main( String args[] )
     {
       Point point = new Point( 72, 115 );
       // get coordinates
       String output = "X coordinate is " + point.getX() + \\
       "\nY coordinate is " + point.getY();
       // set coordinates
       point.setPoint( 10, 10 );
       // use implicit call to point.toString()
       output += "\n\nThe new location of point is " + point;
       JOptionPane.showMessageDialog( null, output, \
       "Point" );
   }
"Demonstrating Class Point",
JOptionPane.INFORMATION_MESSAGE);
System.exit( 0 );
}
} // end class Test

b) Create Circle.java. Point class imports the Point class so we do not have to show the
class definition her again. Circle extends class Point and this means that it includes Point
methods and well as Circle methods setRadius, area, toString and the Circle constructors.
Declare a instance variable (double) protected radius for the radius of the circle. Create
two constructors: One takes in no parameters (no argument constructor) and inside only
has a setRadius(0), and the other constructor takes in double circleRadius and int
xCoordinate, int y Coordinate and inside call super class constructor to set coordinates
“super(xCoor.,yCoor)” and also setRadius(circleRadius). Make method called set
Radius, which passes in double circleRadius and sets it to radius. Make another method
that get radius getRadius(). Finally, make a double area(), which calculates the area of
circle and uses “return Math.PI * radius * radius;”. Finally convert Circle to a String,
public String toString() and it returns “Center = “ + “ [ “ + x + “ , “ + y + “] “ + “; Radius = “ + radius;

GIVEN: TESTCI.java  // to test circle.java
//Test.java
// Applet to test class Circle

// Java core packages
import java.text.DecimalFormat;

// Java extension packages
import javax.swing.JOptionPane;

// Deitel packages
import com.deitel.jhtp4.ch09.Circle;

public class Test {

    // test class Circle
    public static void main( String args[] ) {
        // create a Circle
        Circle circle = new Circle( 2.5, 37, 43 );
        DecimalFormat precision2 = new DecimalFormat( "0.00" );

        // get coordinates and radius
        String output = "X coordinate is " + circle.getX() + "\nY coordinate is " + circle.getY() + "\nRadius is " + circle.getRadius();

    }
}

public class Circle {

    protected double radius;

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

    // setRadius method
    public void setRadius( double radius ) {
        this.radius = radius;
    }

    // getRadius method
    public double getRadius() {
        return radius;
    }

    // area method
    public double area() {
        return Math.PI * radius * radius;
    }

    // toString method
    public String toString() {
        return "Center = [ " + getX() + " , " + getY() + "] ; Radius = " + getRadius();
    }
}

public class Test {
    public static void main( String args[] ) {
        // create a Circle
        Circle circle = new Circle( 2.5, 37, 43 );
        DecimalFormat precision2 = new DecimalFormat( "0.00" );

        // get coordinates and radius
        String output = "X coordinate is " + circle.getX() + "\nY coordinate is " + circle.getY() + "\nRadius is " + circle.getRadius();

    }
}
// set coordinates and radius
circle.setRadius( 4.25 );
circle.setPoint( 2, 2 );

// get String representation of Circle and calculate area
output +=
   "\n\nThe new location and radius of c are\n" + circle +
   "\nArea is " + precision2.format( circle.area() );

JOptionPane.showMessageDialog( null, output,
   "Demonstrating Class Circle",
   JOptionPane.INFORMATION_MESSAGE );

System.exit( 0 );
}

c) Create Cylinder.Java. Cylinder class extends class Circle. This means that it now includes
the methods of Circle and Point. Declare protected double height variable (height of cylinder).
First constructor is a no-argument and setHeight(0)(implicit call to super class constructor here).
The Second constructor takes in 4 parameters cylinderHeight, cylinderRadius, xCoordinate,
yCoordinate. And calls superclass( and takes in 3 parameters cylinderRadius, xCoor, and yCoor
and also calls setHeight( cylinderHeight). Now make a set method called setHeight and it takes
in cylinderHeight as a parameter and inside you equal it to height. Now make a getHeight() and
that returns the height. Make a public double area() and that calculates area of Cylinder: Code
give: “return 2 * super.area() + 2 * Math.PI * radius * height; “ And also calculate volume of
Cylinder in another function called public double volume() Code given: “return super.area() *
height; “.
Finally, convert the Cylinder to a String. Public String toString() and write “return
super.toString() + “; Height = “ + height;

GIVEN: TESTCY.JAVA
// Test.java
// Application to test class Cylinder

// Java core packages
import java.text.DecimalFormat;

// Java extension packages
import javax.swing.JOptionPane;

// Deitel packages
import com.deitel.jhtp4.ch09.Cylinder;

public class Test {

   // test class Cylinder
   public static void main( String args[] )

}
// create Cylinder
Cylinder cylinder = new Cylinder( 5.7, 2.5, 12, 23 );
DecimalFormat precision2 = new DecimalFormat( "0.00" );

// get coordinates, radius and height
String output = "X coordinate is " + cylinder.getX() + " \nY coordinate is " + cylinder.getY() + " \nRadius is " + cylinder.getRadius() + " \nHeight is " + cylinder.getHeight();

// set coordinates, radius and height
setHeight( 10 );
setRadius( 4.25 );
setPoint( 2, 2 );

// get String representation of Cylinder and calculate area and volume
output += "The new location, radius and height of cylinder are: \n\nArea is " + precision2.format( area() ) + " \nVolume is " + precision2.format( volume() );

JOptionPane.showMessageDialog( null, output, "Demonstrating Class Cylinder", JOptionPane.INFORMATION_MESSAGE );

System.exit( 0 );
} // end class Test
a) Point.java class

// Point.java
// Definition of class Point
package com.deitel.jhtp4.ch09;
public class Point {
    protected int x, y; // coordinates of Point
    // No-argument constructor
    public Point() {
        // implicit call to superclass constructor occurs here
        setPoint( 0, 0 );
    }
    // constructor
    public Point( int xCoordinate, int yCoordinate ) {
        // implicit call to superclass constructor occurs here
        setPoint( xCoordinate, yCoordinate );
    }
    // Set x and y coordinates of Point
    public void setPoint( int xCoordinate, int yCoordinate ) {
        x = xCoordinate;
        y = yCoordinate;
    }
    // get x coordinate
    public int getX() {
        return x;
    }
    // get y coordinate
    public int getY() {
        return y;
    }
    // convert into a String representation
    public String toString() {
        return "\[" + x + ", " + y + "]";
    }
}  // end class Point

b) Circle.java class
// Circle.java
// Definition of class Circle
package com.deitel.jhtp4.ch09;

public class Circle extends Point { // inherits from Point
    protected double radius;

    // no-argument constructor
    public Circle() {
        // implicit call to superclass constructor occurs here
        setRadius(0);
    }

    // constructor
    public Circle(double circleRadius, int xCoordinate, int yCoordinate) {
        // call superclass constructor to set coordinates
        super(xCoordinate, yCoordinate);
        // set radius
        setRadius(circleRadius);
    }

    // set radius of Circle
    public void setRadius(double circleRadius) {
        radius = (circleRadius >= 0.0 ? circleRadius : 0.0);
    }

    // get radius of Circle
    public double getRadius() {
        return radius;
    }

    // calculate area of Circle
    public double area() {
        return Math.PI * radius * radius;
    }

    // convert the Circle to a String
    public String toString() {
        return "Center = \[" + x + ", " + y + "]\" + "; Radius = " + radius;
    }
} // end class Circle

c) Cylinder.Java class

// Cylinder.java
// Definition of class Cylinder
package com.deitel.jhtp4.ch09;
public class Cylinder extends Circle {
    protected double height;  // height of Cylinder
    // no-argument constructor
    public Cylinder()
    {
        // implicit call to superclass constructor here
        setHeight( 0 );
    }
    // constructor
    public Cylinder( double cylinderHeight, double cylinderRadius,
            int xCoordinate, int yCoordinate )
    {
        // call superclass constructor to set coordinates/radius
        super( cylinderRadius, xCoordinate, yCoordinate );
        // set cylinder height
        setHeight( cylinderHeight );
    }
    // set height of Cylinder
    public void setHeight( double cylinderHeight )
    {
        height = ( cylinderHeight >= 0 ? cylinderHeight : 0 );
    }
    // get height of Cylinder
    public double getHeight()
    {
        return height;
    }
    // calculate area of Cylinder (i.e., surface area)
    public double area()
    {
        return 2 * super.area() +
                2 * Math.PI * radius * height;
    }
    // calculate volume of Cylinder
    public double volume()
    {
        return super.area() * height;
    }
    // convert the Cylinder to a String
    public String toString()
    {
        return super.toString() + "; Height = " + height;
    }
}
EXPECTED RESULTS
Test.java for Point.java
X coordinate is 72
Y coordinate is 115

The new location of point is [10, 10]

Test.java for Circle.java
X coordinate is 37
Y coordinate is 43
Radius is 2.5

The new location and radius of c are
Center = [2, 2]; Radius = 4.25
Area is 56.75

Test.java for Cylinder.java
X coordinate is 12
Y coordinate is 23
Radius is 2.5
Height is 5.7

The new location and radius and height of cylinder are
Center = [2, 2]; Radius = 4.25; Height = 10.0
Area is 380.53
Volume is 567.45