L07 - Shapes Lab

In this lab, we'll work with Abstract Classes.

1. Start VS Code.
2. During the lecture we discussed the following classes:
   - `Shape` – an abstract class
   - `Circle` – a class that is derived from Shape
   - `Ellipse` – a class that is also derived from Shape
3. Using the `Circle` class a guide and the Ellipse slide, finish writing the `Ellipse` class
4. Using the `Circle` class as a guide, write a class called `Rectangle`:
   - extend the `Shape` class
   - define member variables of type `double` for the `width` and `height`
   - create one or more constructors, as appropriate
   - override the `equals`, `toString`, `findArea`, and `findPerimeter` methods
5. Create a new class with the name `ShapeTest`.
6. This is the body of the class `ShapeTest`:

```
public class ShapeTest {
    public static void main(String[] args) {
        Shape[] myShapes = new Shape[10];
        Circle myCircle = new Circle(10.0);
        double expected = 314.16;
        System.out.println("Circle Area: " + myCircle.findArea());

        myShapes[0] = new Circle(10.0);
        expected = 314.16;
        System.out.println("Circle in Shapes Array Area: " + myShapes[0].findArea());

        myShapes[1] = new Ellipse(5.0, 10.0);
        expected = 157.08;
        System.out.println("Ellipse in Shapes Array Area: " + myShapes[1].findArea());

        myShapes[2] = new Rectangle(5.0, 10.0);
        expected = 50.0;
        System.out.println("Rectangle in Shapes Array Area: " + myShapes[2].findArea());
    }
}
```

When you run the tests, notice that Java knows what particular kind of object (`Circle`, `Rectangle`, or `Ellipse`) has been stored in the array of `Shapes`.

Revision #1
Created 24 April 2025 22:12:03 by Brandon Duke
Updated 24 April 2025 22:12:34 by Brandon Duke