**LAB ASSIGNMENT 4.1**

**Rectangle**

Create a new project called ObjectBehavior. In this new project, create a class called Rectangle.

It is always important to have a well-designed class before writing down any code. Having a class laid out on paper before writing the code allows programmers to see any design flaws before they have coded those flaws into their classes. Determining which classes are needed, what data those classes hold, and how those classes behave are the main objectives of OOP.

The specifications of a class that models a rectangular shape would be:

# Variables

**private double** myX; // the x coordinate of the rectangle

**private double** myY; // the y coordinate of the rectangle

**private double** myWidth; // the width of the rectangle

**private double** myHeight; // the height of the rectangle

**private** DrawingTool pen;

**private** SketchPad paper;

// Creates a 500 x 500 SketchPad with a DrawingTool, pen, that is used

// to display Rectangle objects. The DrawingTool is declared static

// so that multiple Rectangle objects can be drawn on the SketchPad

// at the same time.

# Constructors

// Creates a default instance of a Rectangle object with all dimensions

// set to zero.

Rectangle()

// Creates a new instance of a Rectangle object with the left and right

// edges of the rectangle at x and x + width. The top and bottom edges

// are at y and y + height.

Rectangle(**double** x, **double** y, **double** width, **double** height)

# Methods

// Calculates and returns the perimeter of the rectangle

**public double** getPerimeter()

// Calculates and returns the area of the rectangle

**public double** getArea()

// Draws a new instance of a Rectangle object with the left and right

// edges of the rectangle at x and x + width. The top and bottom edges

// are at y and y + height.

**public void** draw()

**Assignment**:

1. Implement a Rectangle class with the following properties.

a. A Rectangle object is specified in the constructor with the left and right edges of the rectangle at x and x + width. The top and bottom edges are at y and y + height.

b. A method getPerimeter calculates and returns the perimeter of the Rectangle.

c. A method getArea calculates and returns the area of the Rectangle.

d. A method draw displays a new instance of a Rectangle object. Refer to *Handout A1.1 DrawingTool Specifications*, for details on DrawingTool methods.

2. Try your rectangle with both the default constructor and with a constructor that can take the x and y coordinates, the length of the rectangle, and the width. Here are some sample constructor calls:

Rectangle rectA = **new** Rectangle();

Rectangle rectB = **new** Rectangle(0, -80, 400, 160);

Rectangle rectC = **new** Rectangle(100, -100, 20, 300);

3. A circle could be constructed and stored in a very similar manner to a rectangle. Write a class that does the same things as your rectangle class, but modified to apply to a circle. Note: For all values of Pi, you may simply use the value 3.14.