Lab 04: User-Defined Classes

Due date: By the end of your lab session of the week of Monday, September 12th, 2016

Goals

- To learn to create your own classes and define their data members and methods
- To learn to create instances of user-defined classes and call their methods.

Description

In this lab, you will develop some classes that represent different shapes. Each of these shapes have different properties for the shape representation; this will be stored in the data members of the class of the shape. You will implement methods to compute other properties of the shape based on the shape's data members.

Setting up IntelliJ

- Click *Applications* on the top left corner of the screen and then choose *Programming* –> IntelliJ IDEA
- Select *"Create new project"* –> *"Next"* –> *"Next"*
- Browse and select the project location as the directory/folder *"cs180"* that you already created. Project name will automatically take the value *"cs180"*. Do not change it. Then click on *"Finish"*.
- Now you should adjust some settings in IntelliJ.
    - Click on *"File"* –> *"Settings"* –> *"Editor"* –> *"Code Style"*. Set "*Right margin (columns)"* to 120. You can also select *"Wrap when typing reaches right margin"*, and your code will automatically wrap to the next line if there are more than 120 characters in a line of code.
    - Still in *"Settings"*, scroll down to *"Tab size"* and *"Indent"* and set both of these to 4, and click *"Apply"* then *"OK"*
- Right click **cs180** under the Project view and create a new module **lab04**
- Right click on the **src** folder under **lab04** to create the classes: **Circle**, **Triangle**, **Rectangle** and **Driver.**
- Right click in the left margin in the editor and select *Show line numbers.*
- In IntelliJ, your code is automatically saved as you type.
- To run your programs, right click on the class under the src folder and select **Create ClassName** to create a new run configuration for this class. Finally, click the run button, an icon that looks like a green triangle.
- **Remember that classes without a main method are not executable.**

Task 1: Circle

You will create a class that represents a Circle. All circles have a **radius** that can be used to calculate the its **circumference** and **area**. Write a class with a **constructor** and the necessary **methods** and **instance variables** to support this properties.

Steps:

1. Declare the class in the **Circle.java** file

Circle.java

```
/**
 * CS180 - Lab 05
 *
 * Explain briefly the functionality of the program
 *
 * @author Your name, youremail@purdue.edu, lab section
 *
 * @version Date
 */

public class Circle {

}
```

2. Add a **private double** variable that represents the **radius**

Instance Variable

```
// private
private double radius;
```

3. Add a **Constructor** right after the instance variables. Pass in a double variable that represents the radius

Constructor

```
// Constructor
public Circle(double radius) {
    // implementation
}
```

4. Create two **methods**:
    a. **getCircumference()**: Returns the circumference of the circle
    b. **getArea()**: Returns the area of the circle

Methods

```
// Methods
public double getCircumference() {
    // implementation
}

public double getArea() {
    // implementation
}
```

Task 2: Triangle

Create a class that represents an equilateral **Triangle**. In this case, you will need to represent the **base** and use it to calculate the **perimeter** and **area**. Follow the same steps you did to create the Circle:

- Declare the class in the **Triangle.java** file
- Add a **private double** variable that represents the **base**
- Add a **Constructor** right after the instance variables. Pass in a double variable that represents the base
- Create the **methods**:
    - **getPerimeter():** Returns the perimeter of the triangle.
    - **getArea():** Returns the area of the triangle.

Task 3: Rectangle

Create a class that represents a **Rectangle**. In this case, you will need to represent the both the **length and the width** and use them to calculate the **perimeter** and **area**. Follow the same steps you did to create the Circle and Triangle.

Task 4: Driver

Create a **Driver** class to manipulate the Circle, Triangle and Rectangle classes. This class will need a main method to be able to be executed. In the main method, create an instance for each of the shapes. To create them, ask the user to **input the radius** of the circle, **the base** of the triangle and **the length and width** of the rectangle. You can choose the way to receive this input, either using option dialogs or the console. Finally, report to the user the perimeter and area for each of the inputted shapes.

Hints:

- Remember that to create an instance of a class, you need to declare the object and instantiate it using the constructor and the *"new"* keyword.
  Example:
  Given a variable `radius` that holds the input given, we can use
  `Circle c = new Circle(radius);` to create a circle.
- Remember that to call a method of a class, you need to use the reference to the class instance.
  Example: `c.getArea()`

Turning in Your Work

You must turn in your work via the terminal. You will be turning in the "**lab05**" folder that contains all your **.java** files.

```
$ turnin -c cs180=COMMON -p lab(number) lab(number)
```

Rubric

- 10 pts: **Circle** is implemented correctly
- 10 pts: **Triangle** is implemented correctly
- 10 pts: **Rectangle** is implemented correctly
- 20 pts: **Driver** is implemented correctly