Lab 6: OOP Who can do this?

Objectives

- Implementing classes and methods
- Overloading
- Static methods

Implement

1. You want to provide simple mathematical operations for a beginner. Write a class called **SimpleMath** that will have **static methods** for the following operations:
    a. Create a package called **lab6**
    b. Create your class **SimpleMath** under the package **lab6**
    c. Implement an *average* method where you have a parameter n. Read *n* floating point numbers and return the average of them.
    d. Implement another *average* method where you have 3 floating point numbers as the parameters and returns the average of these numbers.
    e. Implement a *max* method where you have a parameter n. Read *n* floating point numbers and return the maximum of them.
    f. Implement another *max* method where you have 3 floating point numbers as the parameters and returns the maximum of these numbers.
    g. Implement a *min* method where you have a parameter n. Read *n* floating point numbers and return the minimum of them.
    h. Implement another *min* method where you have 3 floating point numbers as the parameters and returns the minimum of these numbers.
    i. In the *main* demonstrate how your methods run.
    j. Create another class **SimpleMathDemo** under the package **lab6.** In the *main* demonstrate how your methods run.
2. You want to design a program for geometric shapes. Let's start with a simple one, Rectangle. Later we can add others. The rectangles are typically represented by the *length* and *width*. If the length and width are equal, we can name this rectangle as a *square*. Anyone dealing with the rectangles may want to calculate the *area* and *perimeter*. Write a class for the rectangles as follows:
    a. Create your class **Rectangle** under the package **lab6**
    b. Declare two *private* instance variables, namely *length* and *width.*
    c. Implement 2 constructors: One has no parameter, and one has 2 parameters.
    d. Implement setters/getters(mutators/accessors)
    e. Implement a method for calculating the *area* of the rectangle as length x width and return.
    f. Implement a method for calculating the *perimeter* of the rectangle as 2 x (length + width) and return.
    g. Implement a method for determining the squares. The name of the method will be *isSquare* and it will return true if the *length*=*width*.
    h. In the *main* create instances of *Rectangle* class and demonstrate how your methods run.
    i. Create another class **RectangleDemo** under the package **lab6.** In the *main* demonstrate how your methods run.

How to turn in this lab

Before turning in any program in this class, remember this mantra:

Just because it works does not mean it's good.

Your grade will also come from the following aspects of your code:

- Submission
- Accuracy/correctness
- Readability
- Neatness
- Presentation
- Style
- Testing
- Commenting

For all labs, turn in only an **electronic** version.
Please submit the followings after all labs:

- zip file of your project (the project folder, not just the .java file(s)): zip file name will be your **YourFirstNameLastNameLab6.zip**
- a single pdf file of all your codes (.java), screenshots of your output for each file: pdf file name will be **YourFirstNameLastNameLab6.pdf**

Submit

- **the zip and pdf files**
- after Monday Lab session until Wednesday 8 AM EST
- from the Nexus Lab submission link that will be accessible in Week 2-10

Ask for help if you are having problems!