Lab 14 (October 25 or October 26)

Instructions: Complete the steps below. **Be sure to show your code to one of the lab TAs before you leave, so that you can receive credit for this lab.** You must also upload a copy of all your source code (.java) files to the link on Blackboard by 11:59 PM on Thursday, October 26.

1. Design a `Fan` class that represents a simple window fan. This class contains:

    a. Three `public static` constants ("final" variables) named **SLOW**, **MEDIUM**, and **FAST** with the values 1, 2, and 3, respectively.

    b. A `private int` data field named "speed" that specifies the fan speed (the default is **SLOW**)

    c. A `private boolean` data field named "on" that specifies whether the fan is currently on (the default value is `false`)

    d. A `private double` data field named "radius" that specifies the radius of the fan (the default value is 5)

    e. A `private String` data field named "color" that specifies the color of the fan (the default value is "blue")

    f. Accessor and mutator methods ("getters" and "setters") for all four data fields

    g. A constructor that does not take any arguments, and creates a fan with all default values

    h. A `public` method named `toString()` that does not take any arguments and returns a `String` with a description of the fan. If the fan is on, the method returns the fan speed, color, and radius as one combined `String`. If the fan is off, the method returns the fan color, radius, and the `String` "fan is off" as one combined `String`.

    Draw a UML class diagram for this class and then implement it. Write a test program that creates two `Fan` objects. Set the first fan's speed to maximum, its radius to 10, and its color to yellow, and then turn it on. Set the second fan's speed to medium, its radius to 5, and its color to blue, then turn it off. Display the objects by invoking their `toString()` methods.

2. Design a class named `Stock` that contains:

    a. A `String` data field named “symbol” for the stock's symbol.

    b. A `String` data field named “name” for the stock's name.

    c. A `double` data field named “previousClosingPrice” that stores the stock price for the previous day.

    d. A `double` data field named “currentPrice” that stores the stock price for the current time.

    e. A constructor that creates a stock with a specified symbol and name.

    f. Accessor methods for all data fields.

    g. Mutator methods for `previousClosingPrice` and `currentPrice`.

    h. A method named `changePercent()` that returns the percentage change from `previousClosingPrice` to `currentPrice`.

    Draw a UML class diagram for this class. Implement the class. Write a test program that creates a `Stock` object with the stock symbol “GOOG”, the name “Google Inc”, and a previous closing price of 100. Set the new current price to 90 and display the price-change percentage.

Grading Guidelines: This lab is graded on a scale of 0-3 points, assigned as follows:

0 points: Student is absent or does not appear to have completed any work for the lab

1 point: Student has written one or both programs, but neither program compiles or runs due to errors.

2 points: Student has written one or both programs; only one compiles and runs without errors, or both compile but produce incorrect output.

3 points: Both programs compile and run correctly, without any apparent errors.