CPSC 1181 - Lab 7 [60 marks]

Objectives:

- Design a graphical application for determining statistics about a collection of text
- Build a graphical application that contains text fields, labels, buttons, and event listeners.

Submission:

- Zip up all of the Java files and submit them to D2L prior to the due date. **Due date set in D2L.**
- Submissions that are less than 24 hours late receive a 10% penalty. Submission that are more than 24 hours late will not be accepted.
- Unzipped submissions or submissions containing .class or other unneeded files will be penalized.

Lab Attendance [5 marks]

You must attend lab to receive these marks. If you are late or leave early, you will receive half of these marks. If you come in late and miss attendance, it is your responsibility to ensure that you are marked as present.

Exercise 1

Develop a simple tool for calculating the roots of a quadratic equation. Your finished program should look as similar as possible (or better!) than the sample shown below. Remember a quadratic equation has the form…

$$ax^2 + bx + c$$

The roots are the values of x that result in a value of zero. Use the quadratic equation to calculate these roots. If you need a refresher… https://www.mathsisfun.com/algebra/quadratic-equation.html

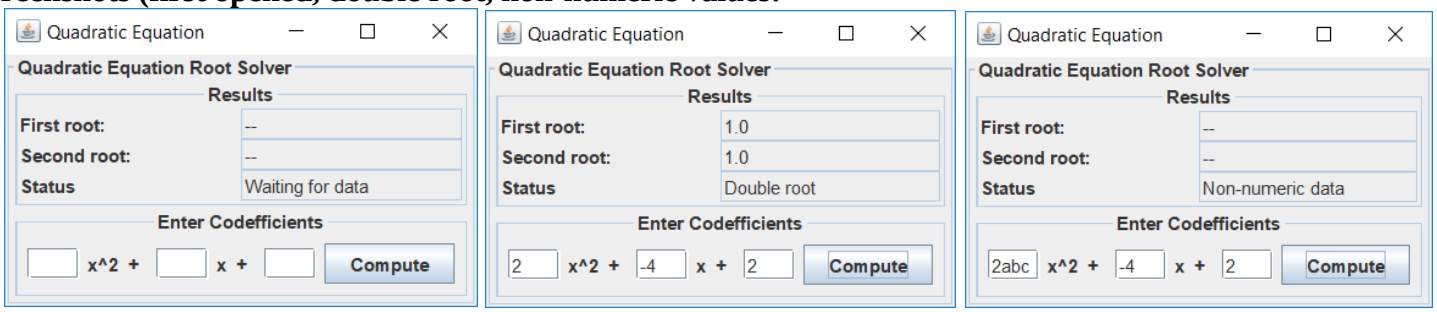
The application should have a single window with two basic areas. The upper area is going to contain the roots of a quadratic equation. The lower area allows the user to enter the coefficients for the equation. You only need to use JLabels, JTextFields, and a JButton. Set the appropriate JTextFields so they can't be edited.

- If the user enters 0 for x^2, then in the status box state that a non-zero value must be selected.
- When opened the status should say 'Waiting for data'. Every time the computation runs, it should be updated with Two unique roots/Double root/Non-numeric data/No real roots or any other needed message.
- If the discriminant is negative (the roots are complex) you don't need to calculate the roots
- In any case where there isn't a root display use – or something similar in the text fields for the roots. Never leave them completely blank.
- The user can enter floating point values for the coefficients
- Display the results with up to 2 decimal places. (5.34, not 5.342573457 or 5.3)
- If the user hits enter after entering a number, the computation should run (don't wait for the compute button to be clicked)
- You can use the following method to handle non-numeric data. It will parse the value and return Double.NaN if the value does not represent a number. (You must still document it)

```
private static double parseDouble(String s) {
    if( s == null ) return Double.NaN;
    try {
        return Double.parseDouble(s);
    } catch (NumberFormatException e) {
        return Double.NaN;
    }
}
```

CPSC 1181 - Lab 7 [60 marks]

Screenshots (first opened, double root, non-numeric values:



Tips:
Remember you can nest panels.
Use layout managers.
Do not place all of your code into the constructor and the event listener. Use good OO thinking to break your program into small pieces. BUT, don't create a separate class for every panel either.
The window will be displayed with small differences on different OSes (Mac vs Windows vs Linux…) Do not worry about minor differences like border color.

Marking Rubric:

Lab Attendance [5 marks]
Style, Convention, Documentation [5 marks]
Good OO Class Design [5 marks]
Main class with Frame [5 marks]

Layout [25 marks]
Functionality [15 marks]