Lab Title: Simple Calculators
Course: CS 1410 – Object Oriented Programming

Objectives: insertion (`<<`) and extraction (`>>`) operators, `cout` and `cin` objects, primitive data types (`int`, `float`, `double`, `bool`, `char`), `const`, `endl`, `fixed`, escape sequences (`\n`, `\t`, `\'`, `\"`), `<iomanip>` library (`setw()`, `setprecision()`), arithmetic operators (`+`, `-`, `*`, `/`, `%`, `++`, `+=`, `--`, `-=`), `<cmath>` library (`pow`).

Overview

In this lab you will build three simple calculators to practice using input and output, variables of primitive data types (int, double, char), basic arithmetic operators, `setw()` and `setprecision()`, and some `cmath` library functions.

For this project, you will need to incorporate the following directives:

```
1   #include <iostream>
2   #include <iomanip>
3   #include <cmath>
4   using namespace std;
```

Program each of the calculators as described in this document. You may incorporate all three calculators into a single `.cpp` file (*not recommended*), or you may elect to use separate files.

Calculator 1: Area of a Circle

Write a simple calculator to compute the area of a circle:

$$A = \pi r^2$$

- You will need two variables of type `double` for `a` and `r`.
- You will need one constant of type `double` for `pi`. Initialize `pi` to 3.1415926535.
- Prompt the user to provide the radius. You will compute and provide the area.
- Use the `setprecision()` function to format the output area to 10 digits; used `fixed` to ensure the same number of digits following the decimal point.

A sample interaction is provided:

```
Radius: 2.75
Area: 23.75829444
```

After you complete the program, run the following test cases. Store your results within the program using comments, as shown.

```
// Test cases:
// Radius          Area
// 1.33             5.5571632448
// 2.75            23.7582944421
// 3.25            33.1830724026
// 4.99            78.2259712314
// 5.50            95.0331777684
```

Calculator 2: Fraction Addition

If there are two fractions, $\frac{a}{b}$ and $\frac{c}{d}$, their sum can be obtained from the formula:

$$\frac{a}{b} + \frac{c}{d} = \frac{a \times d + b \times c}{b \times d}$$

For example, $\frac{1}{3}$ plus $\frac{2}{5}$ is:

$$\frac{1}{3} + \frac{2}{5} = \frac{1 \times 5 + 3 \times 2}{3 \times 5} = \frac{11}{15}$$

Write a program that prompts the user to enter two fractions, using the format: `a/b`, then compute and display the sum in fraction form. For this project, there is no need to simplify or reduce the fractions. A sample interaction is provided:

```
First fraction (a/b): 1/3
Second fraction (c/d): 2/5
Sum: 11/15
```

You will need variables of type `int` for `a`, `b`, `c`, and `d`. You will also need a variable of type `char` that can serve as a placeholder for the forward slash (`/`) in the fractions. The extraction operator (`>>`) can be chained to read in more than one quantity at once, as shown:

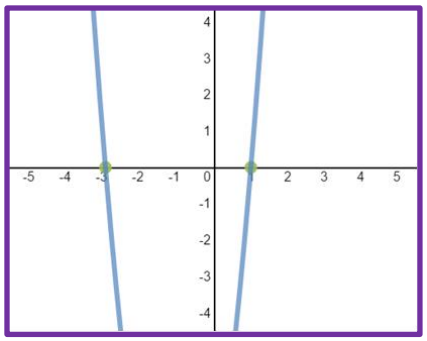
```
cin >> a >> placeholder >> b;
```

After you complete the program, run the following test cases. Store your results within the program using comments, as shown.

```
// Test cases:
// Fraction 1       Fraction 2          Sum
// 1/3              2/5                 11/15
// 1/4              2/3                 11/12
// 1/2              1/2                 4/4
// 7/10             9/15                195/150
// 76/80            95/100              15200/8000
```

Calculator 3: Quadratic Formula

The quadratic formula is used to solve for the values of two x-intercepts, given a quadratic equation. The x-intercept is the value of x where y is equal to zero.



Consider the following equation: $y = 3x^2 + 6x - 9$. A graph of the equation (*right*) reveals x-intercepts at -3 and 1.

Similarly, the equation can be solved by factoring, whereby it is rewritten as $y = 3(x + 3)(x - 1)$.

Some quadratic equations are not easily factored, however, and the quadratic formula may be used to solve them. The quadratic equation is provided:

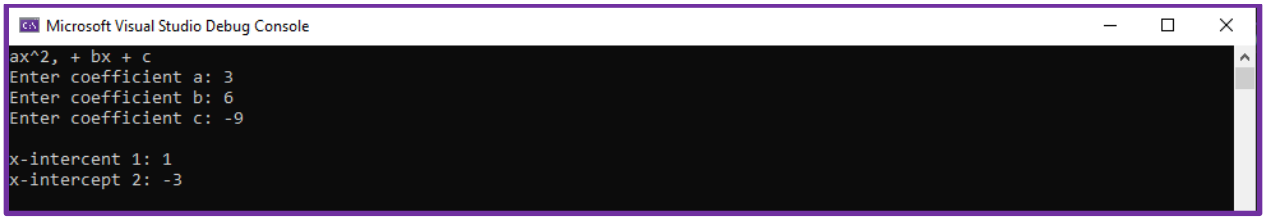
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The quadratic formula is used to solve quadratic equations that are set up in the form $ax^2 + bx + c = 0$, whereby a, b, and c are the numerical coefficients of the equation.

Write a calculator to solve for both versions (+ and -) of the quadratic equation, given coefficients *a*, *b*, and *c*.

- You will need variables for `a`, `b`, `c`.
- Prompt the user to provide the coefficients a, b, and c. You may use separate input statements, or do it all at once with a delimiter (such as a comma).
- Compute both the positive and negative versions of the quadratic formula. Display both to the console.

A sample interaction is provided:



Run and record the following test cases to ensure accuracy:

```
//  Input                   Output
//  a       b       c       x1          x2
//  3       6       -9      1           -3
//  1       -2      -24     6           -4
//  6       3       -6      0.780776    -1.28078
//  2       4       -8      1.23607     -3.23607
//  1       4       -20     2.89898     -6.89898
```

Submit the .cpp file(s) online for grading. A grade rubric for the project is included on next page.