HW: Quadratic Equation Solver

Points

Points	
85	Runs correctly
15	[Code Review](#)
100	**TOTAL**

Submission

1. **Source Code**
   - Submit on Mimir
   - Your program code should be submitted as a single source (.cpp) file

Objectives

- Use selections constructs
  - Do not use the complex number class that you might find. This will prevent you from getting experience using selection statements in C++.

Specifications

Program

The requirements for our solver were updated. We now have to solve the linear solution that can result when $a=0$, and we have to output imaginary roots as well!

Requirements

- **Do not use the complex number class.**
- The program should give all roots including imaginary ones.
- If the coefficients constitute a linear equation you should calculate the single root.
- Input:
  - Three coefficients (a, b, and c respectively) on a single line separated by spaces.
- Output:
  - The equation being solved. See Sample Run below for format.

- ○ Roots.
  - ■ Each root on a separate line.
  - ■ If there is more than one root, then the one obtained by **subtraction** is first.
  - ■ Roots should be reported as
    - ■ `x = <number>`↵
    - ■ `x = <number> - <number>i`↵
    - ■ `x = <number> + <number>i`↵
  - ■ Note that there is a single space ( ) on either side of an '=', a '+' or a '−'.
- ○ If no valid solutions can be calculated, then output:
  `Unable to determine root(s).`↵

- ○ Note that endl puts a line return (↵) at the end.

Hints

- ● You are not required to use the following, but it might be useful!
  - ○ You might need to include <cmath>.
    - ■ To get the square root use sqrt().
    - ■ To get the absolute value
      - ● abs() for ints
      - ● fabs() for doubles

Sample Runs

This is not complete testing!

Note, spacing is important. When working with code and output in a terminal, we try to use a monospaced font such as Source Code Pro (my favorite), courier, or Consolas. They make every character the exact same width. This helps you know exactly how many spaces are there, and work with alignment correctly.
For example:

`Monospaced:`   `Hello`
                  `Water`
`Proportional Font:` Hello
                  Water

- ■ ↵ represents an endl or end of line (You won't actually see that symbol in your output, this is just to make it clear when it is present and when it is not.)

```
1 5 2↵
1x^2 + 5x + 2 = 0↵
x = -4.56155↵
x = -0.438447↵
```

- ```
  3 4 -4↵
  3x^2 + 4x - 4 = 0↵
  x = -2↵
  x = 0.666667↵
  ```

- ```
  -7 9 -8↵
  -7x^2 + 9x - 8 = 0↵
  x = 0.642857 + 0.854161i↵
  x = 0.642857 - 0.854161i↵
  ```

- ```
  3 4 4↵
  3x^2 + 4x + 4 = 0↵
  x = -0.666667 - 0.942809i↵
  x = -0.666667 + 0.942809i↵
  ```

- ```
  3 4 5↵
  3x^2 + 4x + 5 = 0↵
  x = -0.666667 - 1.10554i↵
  x = -0.666667 + 1.10554i↵
  ```

- 0 2 -1↵

  0x^2 + 2x - 1 = 0↵

  x = 0.5↵

- 1 -2 1↵

  1x^2 - 2x + 1 = 0↵

  x = 1↵

- 0 0 3↵

  0x^2 + 0x + 3 = 0↵

  Unable to determine root(s).↵

- 0 0 0↵

  0x^2 + 0x + 0 = 0↵

  Unable to determine root(s).↵

- -3.5 6.2 -9.7↵

  -3.5x^2 + 6.2x - 9.7 = 0↵

  x = 0.885714 + 1.40959i↵

  x = 0.885714 - 1.40959i↵

- -2.3 7.6 -3.1↵

  -2.3x^2 + 7.6x - 3.1 = 0↵

  x = 2.8277↵

  x = 0.476652↵