Approach

Given that this assignment is a continuation, I had to make sure that project 1 was working fully as intended. Therefore, in a way, I had to evaluate and construct 2 programs to complete this assignment. I knew I would be using the Java language given that Project 1’s language is constructed using this language.

Assumptions

1. **A homemade math functions class will be needed –** I was a taught to always make your code look as neat as possible. At first, I knew at the minimum that each math function will have its own method. Then I evaluated the common purpose of these functions and decided to group them together in a class.
2. **Make use of the text file –** Thankfully, building the GUI was easy given that all I had to do was create text in the text file and edit the project to account for the function buttons.
3. **Learn from last project and start early –** Last project I procrastinated, and it almost cost me my project submission. This time around I had to make sure that I avoided that by starting this project early.

Not Implemented

I believe I implemented all the requirements to make this program functional.

Lessons Learned

1. **Creating a calculator –** In the past I have created a calculator as a console app. This is the first time I have created a calculator as a GUI. I was able to learn how calculator UI’s are created by building my own.
2. **Structural thinking –** I have stumbled a few times during the creation of this program, especially in the debugging stage. However, it gave me the opportunity to slow down and think piece by piece. For example, I was stuck on figuring how to evaluate an expression if the user adds or multiplies the result by another number. I was able to use the debugger and concentrate on that specific location in the code rather than trying to evaluate the program as a whole.

Possible Improvements

1. Comment more often and more descriptively.
2. Organizing my code more efficiently.

Table of Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **What aspect is tested** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| Initial execution | Button presses: 5 + 5 \* 2 – 6 / 2 = | Text field result showing 7 | Text field result showing 7 | Pass |
| Evaluating negative numbers | Button presses: 7 \* 2 -/+ = | Text field result showing -14 | Text field result showing -14 | Pass |
| Evaluating expressions attempting to divide by zero | Button presses: 3 4 / 0 | Text field result showing “Cannot Divide by zero” | Text field result showing “Cannot Divide by zero” | Pass |
| Evaluating decimal numbers | 5 . 6 + 7 6 . 2 2 / 11 . 7 | Text field result showing “6.993162393162393” | Text field result showing “6.993162393162393” | Pass |

Screenshots

1.

A close up of a computer

Description automatically generated

2.

A close up of a computer

Description automatically generated

3.

A screenshot of a cell phone

Description automatically generated

4.

A close up of a device

Description automatically generated

­