Dyln Xiong

CSCI 25

Instructor Evan

In this written portion of the assignment, RPN Calculator, I will be going over what I see from the calculator that I have in Github. In this assignment, we were told to build an RPN Calculator by using push, pop, and stacks.

So to start off, I put #include “header.hpp” into my calculator to make it so my header from a different coding space would be included into my calculator. I used a header because it makes it much easier to include a bunch of things while giving itself some room to work with. In my header, I first things I have are #include <stack> and #include <cstdio>. These two codes basically allow my calculator to stack and print out numbers. Without these two my calculator wouldn’t be able to function at all. Next what I have in my header is using std::stack; and using std::printf;. These two codes basically tell my header to use stack and printf inside my calculator. Without these two codes, my calculator wouldn’t use any stacks or printf at all making my calculator not function properly. Lastly, what I have next in my header is int main(); which is the entry point to all my other integers. My integers include int addition (int, int);, int subtraction (int, int);, int multiplication (int, int);, and int division (int, int). These integers basically allow my calculator to add, subtract, multiply, and divide. Without these codes, my calculator wouldn’t be able to add, subtract, multiply, and divide numbers that I put in. So this concludes what is inside my header and now we will be moving on to what is inside my calculator.

Now onto what is in my calculator, like what I mentioned before, first off is my #include “header.hpp”. Next thing I did was put int main( ); which tells my calculator that that is the entry point to start its operations. The next thing I did inside my calculator was build a stack. This stack basically allows my calculator to push numbers and pop out numbers. Without this, my calculator wouldn’t know what to do with the numbers I put it. I also labeled st as stack in my calculator to make writing go by faster and to be more efficient. Inside of my stack was also a generic simple push and pop that told my calculator to add, subtract, multiply, and divide. To make the calculator do the certain algebra, I added if ( ) to each different algebra with an x-value which tells the calculator when to do the certain algebra that I want at the moment. I also implemented a code where if something went wrong it would print out FAIL and if the calculator was functioning right that it would print out the results using printf. Then after all this I added return 0; which tells my calculator that after printing out my results that it is the end of the operation. Lastly, under all of this, I have integers which implement what order to do algebra in. An example of this is, int addition (int l, int r) {return l + r;}, this basically tells the calculator to do algebra from left to right. This concludes what is inside of my calculator. I will now move onto what I got and what I can assume in a logical sense.

So this is what I got from doing this assignment and look at the coding from the calculator. I am very sure that I still don't really understand what I am looking at since I still wouldn't be able to build one by myself unless I got help from someone who actually understood how to use a stack, push, and pop inside of a coding space. I get why we have to push and pop while inside of a stack but at the same time it just doesn't click inside of the coding space. I assume that if we can do simple algebra with just push, pop, and stack. Then if we had more different coding techniques, we would be able to build a proper calculator while having to make a few minor changes to our current calculator. I also assume that making a proper calculator is very tedious as just this simple one is already very much difficult for me to make on my own. This concludes my written portion of the assignment RPN Calculator.