Lab Fb Design

Student Code: Stack with `push(), peek(), pop(),` and `isEmpty().`

```
Class Calculator {
Private:
     Stack calcStack;
     Int i;
     Float num,
          First,
          Second,
          Result;
     String str;
Public:

Int RPNcalc () {
     Cout << Prompt user for a string of ints and operators
     Cin >> string str

     Process string in a for str.length() loop {

          If (string[i] is a number)
               Stack.push(string[i])

          If (string[i] == +)
               First = stack.peek();
               Stack.pop();
               Second = stack.peek();
               Stack.pop();
               Stack.push(first + second)

          If (string[i] == -)
               First = stack.peek();
               Stack.pop();
               Second = stack.peek();
               Stack.pop();
               Stack.push(first - second)

          If (string[i] == *)
               First = stack.peek();
               Stack.pop();
               Second = stack.peek();
               Stack.pop();
               Stack.push(first * second)

          If (string[i] == /)
               First = stack.peek();
               Stack.pop();
               Second = stack.peek();
               Stack.pop();
               Stack.push(first / second)
     }

     //once the loop is done processing the string
     Return stack.peek();
};

Main () {

Bool stop = false;
Char choice = ‘0’;
Calculator calc;

Do {
Calc.RPNcalc();

Cout << Prompt to continue
Cin >> choice;
     Change bool if user wants to quit
} while (!stop)

return 0;
```

<u>TESTING</u>

I first tested the stack to make sure it stored numbers as a first-in last-out structure. Once I saw that it worked as expected (because pushing 10 numbers popped them in reverse order), I moved on to implement the Reverse Polish Notation calculator.

I tried numerous times and various of ways of implementing a do-while loop in my main function, but I was not successful and eventually realized that because that wasn’t a requirement I was wasting a lot of time on something that was not relevant. I hope to implement that into my program in the future. I also deviated from my original thought of treating the string as an array, choosing instead to treat it as a stream object.

I tested my calculator as follows:

Input string	Expected result	Actual result	Match?
5 1 2 + 4 * + 3 -	14	14	Yes
2 5 6 + *	22	22	Yes
8 4 / 5 13 + *	36	36	Yes
4 13 5 / +	6 (6.6)	6	Yes
2 1 + 3 *	9	9	Yes
8 10 5 + 13 - /	4	4	Yes
100 10 / 8 + 2 *	36	36	Yes

With the help of http://www.meta-calculator.com/learning-lab/reverse-polish-notation-calculator.php

After running various tests while writing the code for the program and implementing the tests above, I am confident that the calculator works as expected. I am very pleased with it and enjoyed this lab a lot more than I thought I would when I first read it over.