CPSC 121
Lab 6
Fall 2018
Eric May

```
//Provided with comments in github
struct Calculator {
	long long LHValue;
	long long * RHValue;
	char lastOperator;
};
```

Pointers and Structs

We will be creating a calculator program that uses a struct definition provided to you. Your program will:

1. Declare a Calculator object, and initialize it
    a. 0 for both values and + for last operator will be good
    b. You'll have to initialize the pointer (with "new") before initializing its value!
2. Accepts a line of input that will be read like a calculator
    a. Valid operations - The user is prompted, then, without choosing, just provides one of the below patterns
        i. Full expressions: 3+5, when entered, displays/stores 8 (Used below)
        ii. Partial expressions: +1, when entered, uses the previous results as a starting point. Continuing from the above example, displays/stores 9
        iii. Repeat last operation: =, when entered, uses the previous operation's results with the same operator and Right Hand value. If entered after the above statement, would display/store 10.
            1. You can make this run when no input is provided, just document it and let me know as the user!
        iv. Quit: q, when entered, ends the execution of the program
    b. Apart from any whitespace that may be in the input string, any characters that are neither digits, an operator (+-*/%), or the quit command, should not be accepted; meaning the calculator command is rejected with a message given to the user. If the user's input is rejected, no action should be taken.
3. Based on the user's valid input, appropriately calculate, display, and store the answer in the LHValue member of your Calculator object
    a. void processInput(Calculator * calc, string userInput)
    b. Function definition above must be used (argument types can't be changed)
4. Repeat step 2

Note: Don't allow for division by 0!

If you're unfamiliar with pointers, wait, look to the slides and/or try working on this assignment with the pointer temporarily converted to an ordinary variable.

Points:
1 - Documentation, readability, format
1.5 - Proper use of Pointers
1.5 - Proper use of Structs
2 - Proper program flow (conditionals, loops, etc)
2 - Filename and Header
2 - Output testing

Header
//Author: Eric May (your name)
//CPSC 121 Lab 6
//<MM/DD/YY> (Current Date)

Filename
<Last Name><First Initial>lab6.cpp
For example, my assignment would be named MayElab6.cpp