CSCI 1380 **Lab 3: Decision Making Control Structure** **Mr. Alfredo Ramos**

Due date: ***On Blackboard***

Problem:

In your own words, analyze and explain the program you implemented below, using just one or two sentences, what it does. Write this explanation as a comment at the top of your program. **This explanation is part of your grade (Look at the rubric)**. Do not be too detailed and/or vague (For example, do not write: it opens a file, then declares variables, and then reads from a file…); it should look like the problem descriptions in my previous lab assignments.

Requirements:

Implement in C++ the algorithm solution (Pseudo-code) shown below:

```
01)  Include your libraries here. Explain the libraries being used with comments.
02)  Declare variable(s) named "numA" and "numB" that hold(s) double-precision real number(s).
03)  Declare variable(s) named "resultA" that hold(s) double-precision real number(s) and initialize it to "0.0".
04)  Declare variable(s) named "opSign" that hold(s) a character(s).
05)  Declare variable(s) named "errorCode" that hold(s) integer number(s) and assign it the value: 0.
06)  Prompt the user:
         Please enter an expression (for example, 2 + 5):
07)  Get the value(s) from the keyboard and store them in the appropriate variable(s).
08)  Check the operator and preform the appropriate arithmetic as follows: [i]
         If operator is equivalent to addition, then
             01)  Calculate the expression: numA + numB and store in appropriate variable.
         Otherwise, if operator is equivalent to subtraction, then
             01)  Calculate the expression: numA - numB and store in appropriate variable.
         Otherwise, if operator is equivalent to multiplication, then
             01)  Calculate the expression: numA * numB and store in appropriate variable.
         Otherwise, if operator is equivalent to division, then
             If divisor is equivalent to zero, then
                 01)  Assign integer error code one (1) to corresponding variable(s)
             Otherwise,
                 01)  Calculate the expression: numA / numB and store in appropriate variable.
         otherwise,
             01)  Assign integer error code two (2) to corresponding variable(s)
09)  Display the following messages based on the error code on the screen: [ii] [iii]
         If error code is equivalent to zero (1), then
             01)  Display on the screen:
                      Illegal operation: Division by zero...
         Otherwise, if error code is equivalent to one (2), then
             01)  Display on the screen:
                      Invalid expression and/or operation...
         Otherwise,
             01)  Display on the screen:
                      The result of [numA] [opSign] [numB] is [resultA].
10)  Display on the screen: [iv]


         Press ENTER key to continue...
 11)  Terminate your program with integer 0.
```

Note: You must use the tools learned in the class, slides, and book from chapter 4 and below; otherwise, you will receive no credit for the lab!

Examples:

CSCI 1380 **Lab 3: Decision Making Control Structure** **Mr. Alfredo Ramos**

Due date: *On Blackboard*

Check out the program samples I provided with the zip file or look at the end for the screenshots of sample test runs you can try. Make sure that your output is the same as mine.

The program **must compile without errors or warnings.**
Your program must include the algorithm solution as comments (implement in C++ each step right below its corresponding comment).

Your program must have the following comments at the top. Don't forget to include them because they will count toward the grade of this lab.

```
//****************************************************************************
// Course Number and Section:
// Course Semester:
// Your Name:
// --Only if he helped you with the assignment--
// Teammate Name:
// ---------------------------------------------
// Program Description:
//
//
//
//****************************************************************************
```

Submission:

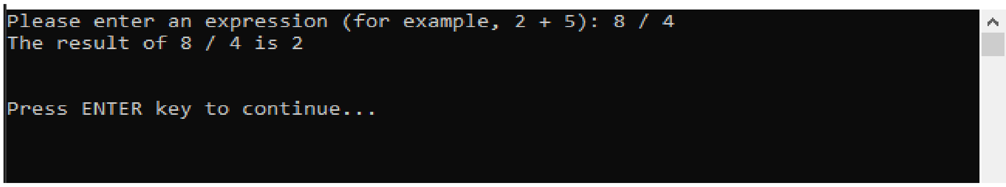
Please name your file **lab3TXX** (where **XX** is the team/group number). If you work with a teammate, then make sure to include them on the lab heading.

DO NOT include blank spaces in the name of the file please.

For example: lab01T00.cpp

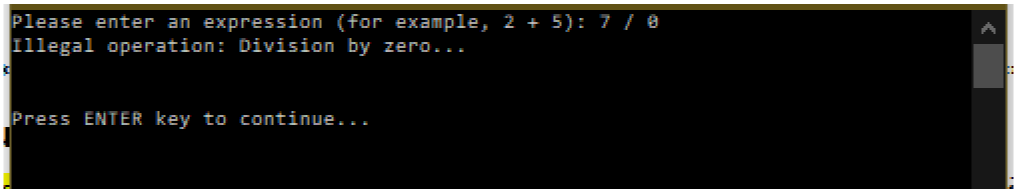
When done, submit your solution through Blackboard using the "Assignments" tool. Do Not email it.

Screenshots:



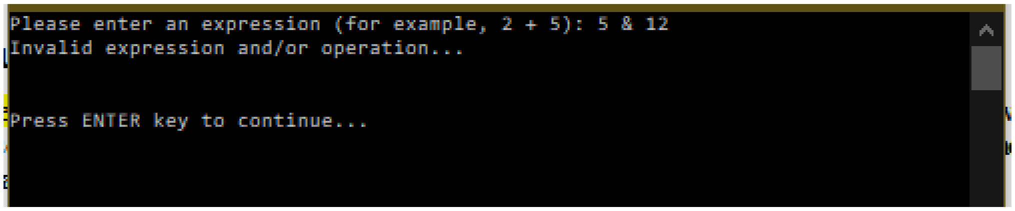
```
Please enter an expression (for example, 2 + 5): 8 / 4
The result of 8 / 4 is 2


Press ENTER key to continue...
```



```
Please enter an expression (for example, 2 + 5): 7 / 0
Illegal operation: Division by zero...


Press ENTER key to continue...
```



```
Please enter an expression (for example, 2 + 5): 5 & 12
Invalid expression and/or operation...


Press ENTER key to continue...
```

Grading Criteria: [v]

	Levels of Achievement			
Criteria	**Exceptional**	**Proficient**	**Satisfactory**	**Unsatisfactory**
Heading **Weight 10.00%**	**100.00%** Included program heading and information is fully completed.	**50.00%** Included program heading, but is missing information.	**25.00%** Included program heading, but did not fill information.	**0.00%** Did not include program heading.
Documentation **Weight 15.00%**	**100.00%** Program is fully commented, comments are placed above the statements and comments are your own words.	**50.00%** More than half comments added to program, comments are placed above the statements, and/or comments are your own words.	**25.00%** Less than half comments added to program, comments not placed above statements and/or comments are not your own words.	**0.00%** No comments added to program.
Organization **Weight 20.00%**	**100.00%** White space is applied appropriately and program follows a structured organization. Ex: Follows algorithm and/or some pattern.	**50.00%** White space is mostly added and follows mostly structured organization. Ex: Follows algorithm and/or some pattern.	**25.00%** White space is minimally added and follows minimally structured organization. Ex: Follows algorithm and/or pattern.	**0.00%** No white space is added and follows no structured organization. Ex: Follows algorithm and/or pattern.
Requirements **Weight 35.00%**	**100.00%** Program requirements were fully met. Look at program handout requirements.	**50.00%** More than half of the program requirements were met. Look at program handout requirements.	**25.00%** Less than half of the program requirements were met. Look at program handout requirements.	**0.00%** Program requirements were not met. Look at program handout requirements.
Accuracy **Weight 20.00%**	**100.00%** Program output is accurate and is formatted appropriately. Ex: Spacing, new lines, etc.	**50.00%** Program output is accurate and/or mostly formatted appropriately. Ex: Spacing, new lines, etc.	**25.00%** Program output is not accurate and/or mostly formatted appropriately. Ex: Spacing, new lines, etc.	**0.00%** Program output is not accurate and is not formatted appropriately. Ex: Spacing, new lines, etc.

[i] Use a nested decision-making structure (if, if-else, if-else-if).

[ii] Use a switch structure (Switch-Case).

[iii] Careful on division expressions. You need to check for a specific condition that is illegal such as division by 0.

[iv] Pay attention to the new lines when displaying the message. Look at screenshot and/or sample executable program.

[v] More points may be lost for other reasons not listed in the criteria rubric and/or requirements.