CSCI 1380 **Lab 4: Decision and Repetition** **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 explination 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 "grade" and "counter" that hold(s) integer number(s). Also, declare
        variable(s) named "total" that hold(s) integer number(s) assigned with the value: -1.
03)     Declare variable(s) named "sum" and "totalPass" that hold(s) integer number(s) and initialize it to
        value(s): 0.
04)     Declare variable(s) named "avg" that hold(s) double-precision real number(s) and initialize to
        value(s): 0.0.
05)     Declare input file stream variable(s) named "inFileA".
06)     Associate the input file stream variable(s) with the name of the file(s) to be used appropriately.
07)     Check if the input file was properly associated and opened. [i]
                If the file was not opened, then
                        01)     Display on the screen: [ii]
                                        Error opening the file...

                                        Press any key to continue...
                        02)     Close the program with integer -1.
08)     While process value(s) is not between (inclusion) zero (0) and hundred (100), then [iii]
                01)     Prompt the user:
                                How many grades do you want to process:
                02)     Get the value from the keyboard and store it in corresponding
                        variable.
                03)     If the input stream variable fails, then
                                01)     Clear the status flag of input stream variable.
                                02)     Clear the input stream variable data (Look at XX slide
                                        in XX PowerPoint).
                                03)     Assign total the value negative one (-1)
09)     While counter is less than process value(s), then [iv]
                01)     Get the grade(s) from the file(s) and store it in the appropriate
                        variable(s).
                02)     If grade(s) is greater than or equal to seventy (70), then
                                01)     Accumulate the grade(s) in the appropriate variable.
                                02)     Increment by one (1) the total passing grade variable.
10)     If process value is not equal to zero (0), then
                01)     Calculate the average and store it in the appropriate variable
                                Formula: sum_of_grades(s) / total_of_passing_grade(s)
        Otherwise
                01)     Display on screen: [v]
                                No values to process...

                                Press any key to continue...
```

    02) Close your program with integer 0.

11) Round your answer to the nearest tenth (One decimal place). [vi]
12) Format output in fixed decimal notation displaying two digit(s) after the decimal point.
13) Display on the screen the message: [vii]

```
The average of [totalPass] passing grade(s) only

out of [total] grades is [avg].


Press any key to continue...
```

14) Close the input stream variable.
15) Terminate your program with integer 0

Note: You must use the tools learned in the class, slides, and book from chapter 5 and below; otherwise, you will receive no credit for the lab!

Examples:
Run my sample solutions to see what your programs should do. Test them with the same value (for example 15) and pay attention to the outputs. Then, determine the averages with a calculator and compare these results with those returned by the programs.

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.

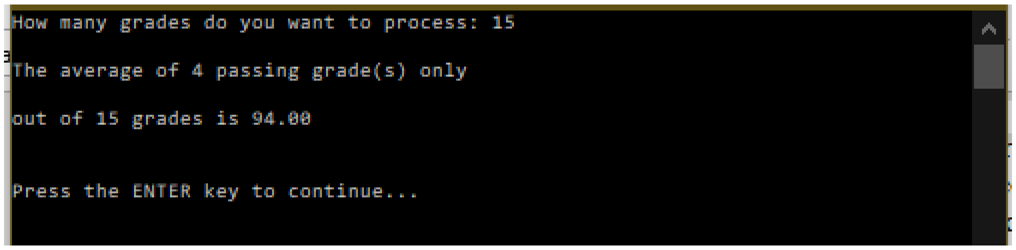
```
//****************************************************************************
// Course Number and Section:
// Course Semester:
// Your Name:
// --Only if he helped you with the assignment--
// Teammate Name:
// ---------------------------------------------
// Program Description:
//
//
//
//******************************************************************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

Submission:

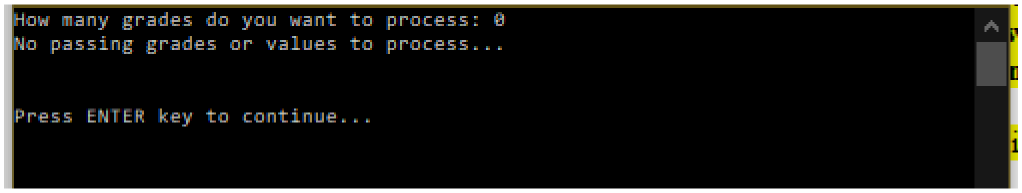
Please name your file **lab4TXX** (where **XX** is the team number). **Do not include blank spaces in the name of the file please.**

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

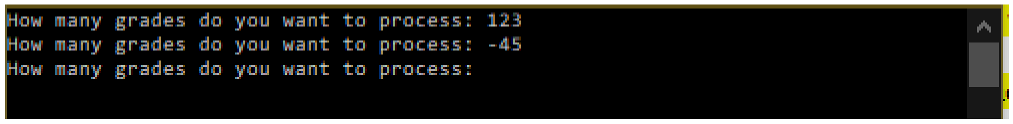
Screenshots:



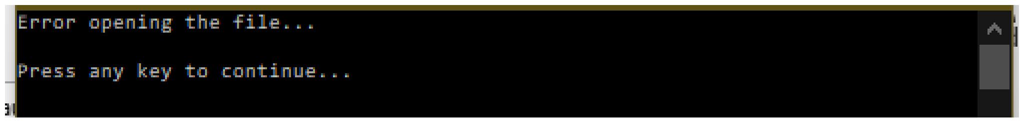
```
How many grades do you want to process: 15

The average of 4 passing grade(s) only

out of 15 grades is 94.00


Press the ENTER key to continue...
```



```
How many grades do you want to process: 0
No passing grades or values to process...


Press ENTER key to continue...
```



```
How many grades do you want to process: 123
How many grades do you want to process: -45
How many grades do you want to process:
```



```
Error opening the file...

Press any key to continue...
```

Grading Criteria:

	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	Look at Chapter 03 for file manipulation.
ii	Pay attention to the new lines above the message
iii	Use a do-while structure for this part of the program.
iv	Use a for-loop structure for this part of the program.
v	Pay attention to the new lines when displaying the message. Look at screenshot and/or sample executable program.
vi	Use the method learned from Chapter 02 for method/algorithm of rounding.
vii	Pay attention to the new lines when displaying the message. Look at screenshot and/or sample executable program.