Objective

To practice and learn the following coding skills:

- Methods with passing a parameter as pass-by-reference **out**
- If-else statements
- Combining Boolean expressions (conditions) with AND (&&) , OR (||)
- Using TryParse() method instead of Parse
- Using bool data type variables

Instructions

We would like to modify the program in assignment3 and add user input validation and more features to it.

Step 1: we must change the method GetUserInput

The method must get 3 inputs and return 2 outputs.

3 inputs are: textMessage, min, and max.

The 2 outputs are:

- the value that user has entered
- a Boolean value that signals if the user input was valid or not (true or false).

Because our method has two outputs, so we consider one of them as a return value (using return statement in the method) and one output as **out** pass-by-reference type parameter. Therefore, GetUserInput method has 4 parameters (one of the them is **out** type) and one return value.

So, the method definition (declaration) must be like:

```
static bool GetUserInput(string textMessage, byte min, byte max, out float userInputValue)
{
    // your code goes here
}
```

min and max parameters are lower limit and upper limit for validity of the user input. For example, valid grade values are from 0 to 100, so min is 0 and max is 100, any value out of this range is invalid.

One example of calling GetUserInput is like:

```
bool isInputValid;
isInputValid = GetUserInput("Assignments", 0, 100, out assignments);
```

Note: although in this assignment, min and max are only 0 and 100 respectively, however, we must write the method such that it works for all different min and max values. In fact, we must write a method such that it works in all our projects without changing the code in the body of the method. Imagine you are working on another assignment and in that assignment the user input is valid if it is a number between 120 and 150. Ask yourself whether your method works if you copy it to the new assignment without changing any line of the code of the GetUserInput method? Does your GetUserInput method works in another project about calendar and months, if you call it as follows?

```
isInputValid = GetUserInput("Number of Month", 1, 12, out monthNumber);
```

Note: in calendar project, valid input is a number between 1 and 12 and if user enters any number out of this range, GetUserInput method must return false.
You can test your method with parameters like example below (min = 20 and max = 50) and test if you enter in Console 60 as input, GetUserInput method must return false, if the user enters 30, GetUserInput method must return true. Use debugger to make sure GetUserInput method works correctly with all different input parameters.

```
isInputValid = GetUserInput("Test", 20, 50, out userValue)
```

Step2: we must change the body of GetUserInput method. A pseudocode[1] for the method body follows:

```
Display a message asking the user to enter a value
Use TryParse method to convert the user input string to a floating-point number value
If the user input is not valid (i.e. it is not a numeric value)
    Display a message (check the executable file provided along with this assignment for the exact message. Note: min and max are not always 0 and 100. Do not forget that we may use GetUserInput in other projects and min and max may be 1 and 12, so the message must display whatever min and max value is, and NOT ALWAYS 0 and 100 )
    Set userInputValue to zero
    Return false value
Else
    If the user entered a value that it is not between min and max inclusive
        Display a message (check the executable file provided along with this assignment for the exact message. Note: min and max are different in different projects)
        Set userInputValue to zero
        Return false value
    Else
        Set userInputValue to the value that the user has entered
        Return true value
```

Step 3: The Main method of assignment 3 must be changed because now if the value for grades are valid (they are numeric and between 0 and 100) then calculation of Weighted Average Grade must be done, else no calculation must be done. (if user's input is invalid, why should we waste processor time?)

There are two solutions for this step. The pseudocodes would be as:

```
If the value that user has entered for Assignments AND Midterm Exam AND Quiz1 AND… are valid
{
        Do calculations (lines 46 to 75 of assignment 3 code)
}
Write("\nPress a key to quit...");
 ReadKey();
```

Or, you can write the same logic differently as:

```
If the value that user has entered for Assignments is valid
        If the value that user has entered for Midterm Exam is valid
                If the value that user has entered for Quiz1 is valid
                        If the value that user has entered for Quiz2 is valid
                                If the value that user has entered for FinalExam is valid
                                        {
                                        Do calculations (lines 46 to 75 of assignment 3 code)
                                        }
Write("\nPress a key to quit...");
 ReadKey();
```

Hint: if a method like MyMethod returns a boolean value, you can write an if-statement like:

```
If ( MyMethod() ){
}
```

[1] Pseudocode is an informal high-level description of the operating principle of a computer program or other algorithm.

Step 4: you must write a new method called LetterGrade. This method has 1 input and 1 return value. This method gets grade number value and returns a letter value (F, D, C-,C, C+, B-,B,B+,A-,A,A+). Input data type is float and return type is string. (for letter grade follow Douglas College Letter Grade Policy)

Hint: write one if statement, nine else-if statement and one else statement. Can you write the method using switch-case? If you can, write a switch-case version too.

Step 5: Change DisplayTableRow method. The method must have 4 inputs instead of 3. The new input parameter is the corresponding letter grade of a numeric grade. One example of calling this method is like:

```
DisplayTableRow("Assignments", ASSIGNMENTS_PERCENTAGE, assignments, LetterGrade(assignments));
```

Step 6: Change the body of one of the two DisplayTableRow methods to incorporate the new change (to display the letter grade)

Step 7: Change the two lines of code where *totalWeightedAverage* and *weightedExams* values are displayed to display letter grades beside the number grade (check my executable file if this step is not clear to you)

Step 8: using an if-else statement, check if the value of *weightedExams* is greater than or equal to 50, display on the console that the student passes the course, otherwise the student fails the course.

The solution to assignment 3 code is provided to you. Apply the changes that are requested in this assignment to it.

You are provided with an executable file for your test. Make sure your program behaves just like this executable file of the assignment.

Requirements

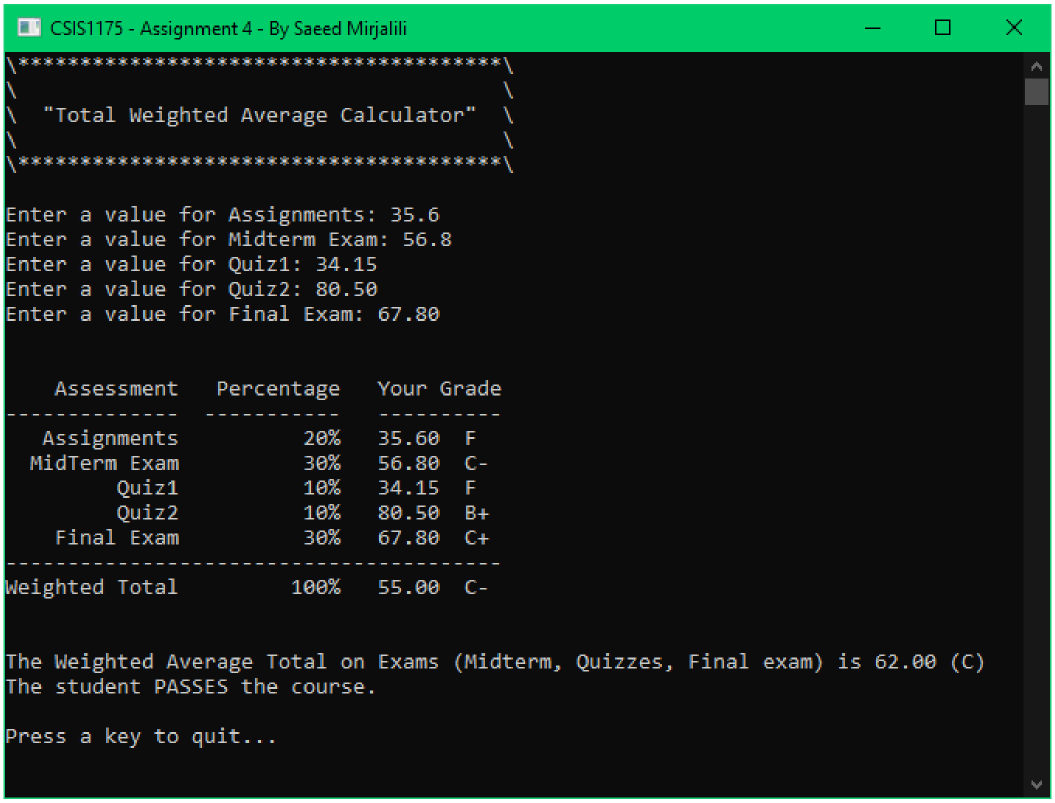
1. On very top of your program, before using directives, write your name, date and purpose as a multiline comment.
2. Make sure your code looks neat and tidy. Press Ctrl+KD in Visual Studio to format your code based on coding styles.
3. Respect Naming Conventions for constants (ALL_CAPS) , variables (camelCase), Method names (PascalCase).

Submissions

Please submit the following items to the corresponding assignment folder on BlackBoard prior to the deadline:

- A zip/archive folder containing the project folder specially source code files (.cs file). Rename the zip file name as a#_xy.zip where # is assignment number, x is your first name and y is your last name. For example, when I submit my work for assignment 2, I will rename the zip file as a2_SaeedMirjalili.zip
- If you do not know how to zip a file or folder, search Google for it (for example click here)

One execution screenshot follows:



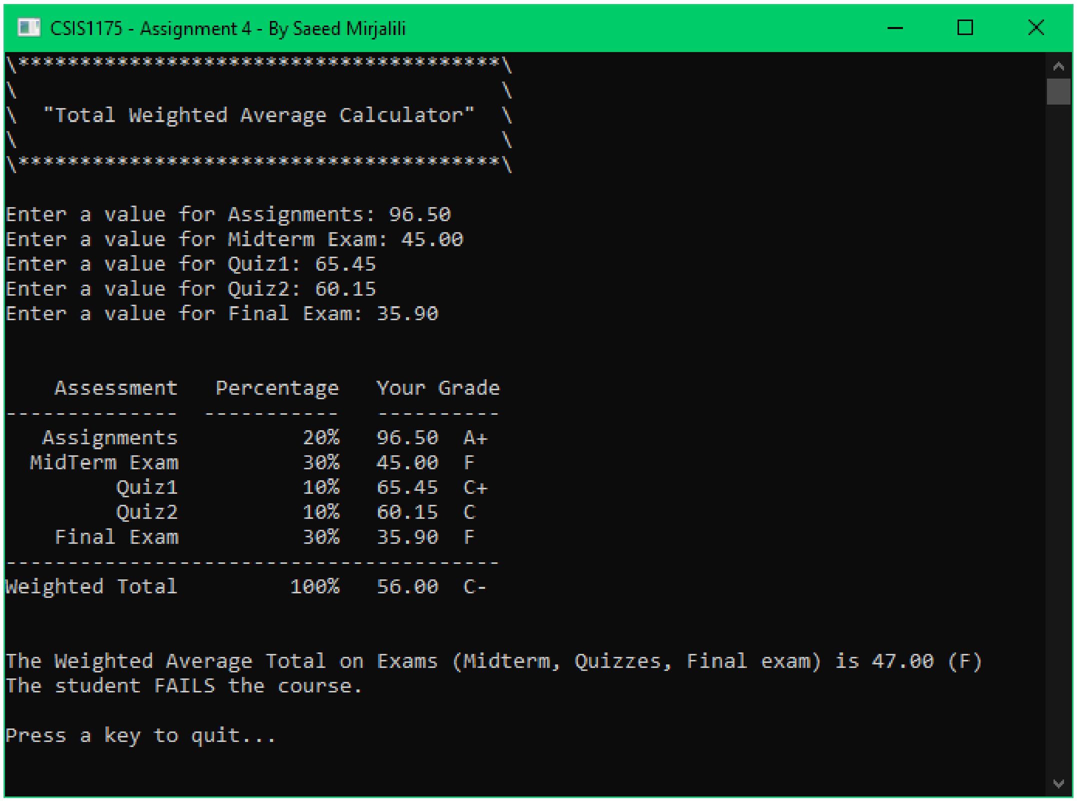
```
\***************************************\
\                                       \
\  "Total Weighted Average Calculator"  \
\                                       \
\***************************************\

Enter a value for Assignments: 35.6
Enter a value for Midterm Exam: 56.8
Enter a value for Quiz1: 34.15
Enter a value for Quiz2: 80.50
Enter a value for Final Exam: 67.80


    Assessment   Percentage   Your Grade
--------------  -----------   ----------
   Assignments          20%   35.60  F
  MidTerm Exam          30%   56.80  C-
         Quiz1          10%   34.15  F
         Quiz2          10%   80.50  B+
    Final Exam          30%   67.80  C+
----------------------------------------
Weighted Total         100%   55.00  C-


The Weighted Average Total on Exams (Midterm, Quizzes, Final exam) is 62.00 (C)
The student PASSES the course.

Press a key to quit...
```



```
\***************************************\
\                                       \
\  "Total Weighted Average Calculator"  \
\                                       \
\***************************************\

Enter a value for Assignments: 96.50
Enter a value for Midterm Exam: 45.00
Enter a value for Quiz1: 65.45
Enter a value for Quiz2: 60.15
Enter a value for Final Exam: 35.90


    Assessment   Percentage   Your Grade
--------------  -----------   ----------
   Assignments          20%   96.50  A+
  MidTerm Exam          30%   45.00  F
         Quiz1          10%   65.45  C+
         Quiz2          10%   60.15  C
    Final Exam          30%   35.90  F
----------------------------------------
Weighted Total         100%   56.00  C-


The Weighted Average Total on Exams (Midterm, Quizzes, Final exam) is 47.00 (F)
The student FAILS the course.

Press a key to quit...
```