

Objectives

To practice and learn the following coding skills:

- Declare memory locations for data (defining variables)
- Use predefined data types (int, float, double, decimal...) efficiently (proper memory size and type)
- Declare and manipulate strings
- Work with constants
- Write expressions using arithmetic operators
- Learn special formatting rules for numbers (integral and floating point) and percentage
- Naming Convention for constants (ALL_CAPS) and variables (camelCase)

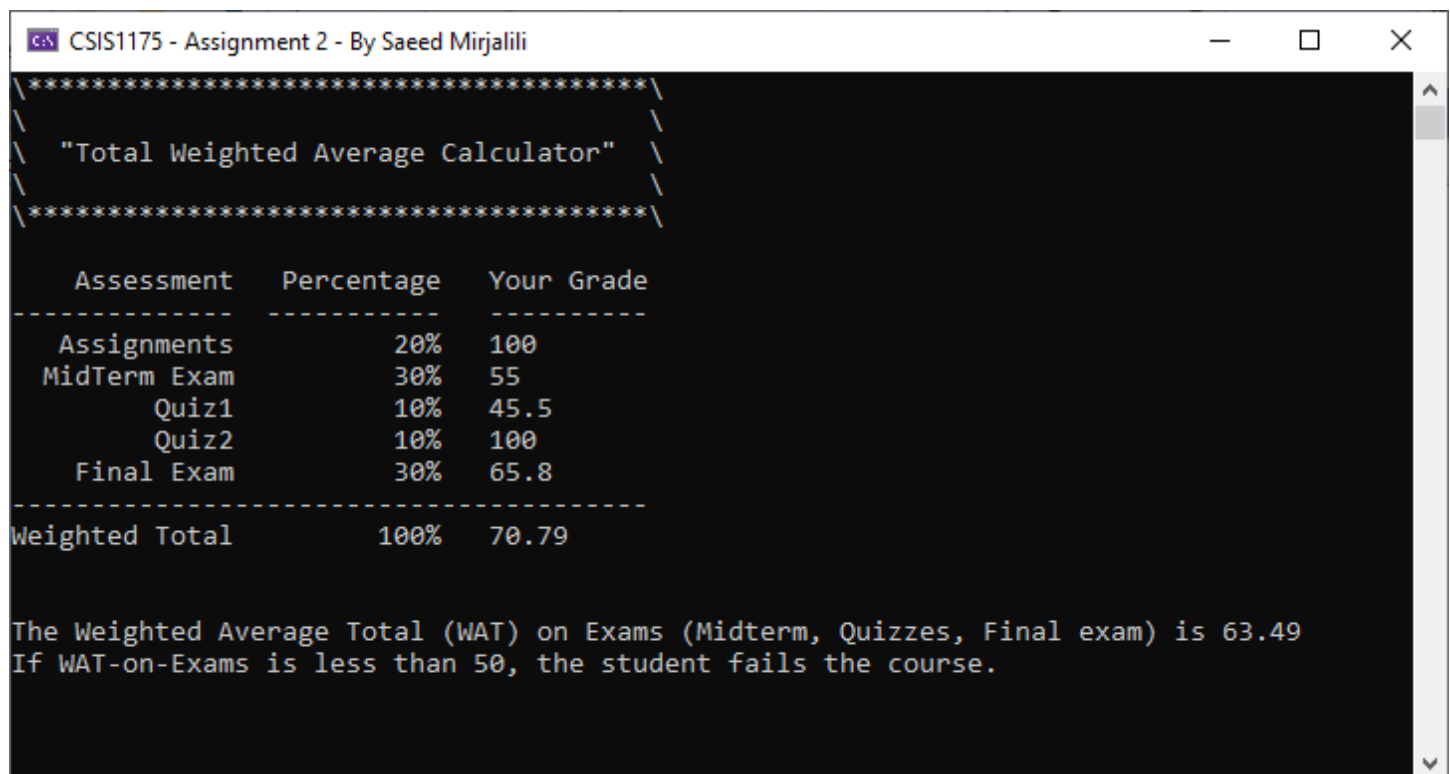
Instructions

Write a program that computes a weighted average (read endnote about weighted average)¹ of final grades of students in 1175, giving the following weights:

- Assignments: 20%
- Midterm exam: 30%
- Quizzes: 20%
- Final Exam: 30%

Do a compile-time initialization (initialize variables and constants in your source code. It means, the user of your program will not type in the input values, instead you as a programmer set initial values to the variables).

You are provided with the following screenshots of several test executions of the program. You can test your program with values that you can see in the screenshots.



```
C:\> CSIS1175 - Assignment 2 - By Saeed Mirjalili

*****\
/"Total Weighted Average Calculator"/\
*****\

Assessment    Percentage    Your Grade
-----
Assignments      20%         100
MidTerm Exam     30%          55
    Quiz1        10%         45.5
    Quiz2        10%         100
    Final Exam   30%         65.8
-----
Weighted Total    100%        70.79

The Weighted Average Total (WAT) on Exams (Midterm, Quizzes, Final exam) is 63.49
If WAT-on-Exams is less than 50, the student fails the course.
```

```

CSIS1175 - Assignment 2 - By Saeed Mirjalili
*****
"Total Weighted Average Calculator"
*****
Assessment    Percentage    Your Grade
-----
Assignments      20%      90
MidTerm Exam    30%      65
Quiz1           10%      60
Quiz2           10%      75
Final Exam      30%      50
-----
Weighted Total      100%    66.00

The Weighted Average Total on Exams (Midterm, Quizzes, Final exam) is 60.00
If WAT-on-Exams is less than 50, the student fails the course.

```

```

CSIS1175 - Assignment 2 - By Saeed Mirjalili
*****
"Total Weighted Average Calculator"
*****
Assessment    Percentage    Your Grade
-----
Assignments      20%      95
MidTerm Exam    30%      65
Quiz1           10%      80
Quiz2           10%      95
Final Exam      30%      60
-----
Weighted Total      100%    74.00

The Weighted Average Total on Exams (Midterm, Quizzes, Final exam) is 68.75
If WAT-on-Exams is less than 50, the student fails the course.

```

```

CSIS1175 - Assignment 2 - By Saeed Mirjalili
*****
"Total Weighted Average Calculator"
*****
Assessment    Percentage    Your Grade
-----
Assignments      20%      80
MidTerm Exam    30%      86
Quiz1           10%      90
Quiz2           10%      95
Final Exam      30%      78
-----
Weighted Total      100%    83.70

The Weighted Average Total on Exams (Midterm, Quizzes, Final exam) is 84.63
If WAT-on-Exams is less than 50, the student fails the course.

```

```

CSIS1175 - Assignment 2 - By Saeed Mirjalili
*****
"Total Weighted Average Calculator"
*****
Assessment    Percentage    Your Grade
-----
Assignments      20%     100
MidTerm Exam    30%       0
Quiz1           10%     45.5
Quiz2           10%     100
Final Exam      30%     65.8
-----
Weighted Total      100%    54.29

The Weighted Average Total (WAT) on Exams (Midterm, Quizzes, Final exam) is 42.86
If WAT-on-Exams is less than 50, the student fails the course.

```

Requirements

- On very top of your program, before **using** directives, write a multiline comment and write the followings:
 - Programmer: (your name)
 - Date: (Fall 2020)
 - Purpose: (explain the purpose of the program)
- Inside the Main() Method, start declaring all constants and variables. Define 4 constants for the percentages of 4 components of the course. Define one percentage for quizzes (20%) and not two percentages for quiz1 (10%) and quiz2 (10%). That is, **one** constant for quizzes percentage and **not two** constants for quiz1 percentage and quiz2 percentage separately.
- Respect naming convention for constants. (**Hint**: There are 3 naming conventions: PascalCase, camelCase, ALL_CAPS. Which naming convention is used for constants? Which one is used for variables?)
- Choose float as the data type for percentages, because percentages are not big numbers.
- Initialize values for percentages. For example: `const float ASSIGNMENTS_PERCENTAGE = 0.2f;` (f is for enforcing the data type to *float*, otherwise the data type is *double* by default. Remove f after 0.2 to see what happens)
- Define variables to store grades of a student for each assessment (assignment, midterm exam, quiz1...) and initializes them with some values out of 100. Choose the most efficient data type for these variables (efficient in terms of size and type (integral or floating point)). Should the type of variable be float? Should be int? long? Choose the most efficient one!
- Use Write() in your program to display on console a box exactly as you can see on the screenshots. Pay attention to the backslash signs at the beginning and ends of every line and double quotation marks around **Total Weighted Average Calculator**. Use only Console.WriteLine() to show this box. **DO NOT USE WriteLine()**. To display this whole box, use only Write() and escape characters (\t, \n, \", \\) to display spaces, new lines, quotations, and backslashes. Use \t instead of many spaces!
- To display the table as you can see on the screenshots, you must use WriteLine() and Format Specifiers (read textbook, pages 110 to 115 or lecture slides chapter2 - #66 onward). For example, to display the row for assignments use a statement like the following:

```
WriteLine("{0,14}{1,13:P0}    {2,-15}", "Assignments", ASSIGNMENTS_PERCENTAGE, assignments);
```

Explanation:

- {0,14} means the first parameter (the string "Assignments"), reserves 14-character column and it would be right aligned.
 - {1,13:P0} means the second parameter (the constant ASSIGNMENTS_PERCENTAGE), reserves 13-character column, and it would be right aligned and displays the value of the parameter (ASSIGNMENTS_PERCENTAGE) as percentage with 0 digits on the fraction part of the number. The percentage sign (%) is automatically added to numbers.
 - {2,-15} means the third parameter (the value of assignments variable) is displayed in a 15-character reserved column, and left aligned (negative means left-aligned).
9. Make sure to test your program with the 5 sets of numbers as you can see on 5 screenshots included in this document and your program must display the same results. To test your code with 5 sets of values (or inputs), you must have 5 copies of all variables in your code with different values. Every time you want to test your program with one set of variables, comment the other 4 sets of variables and **uncomment** only one set of variables. This way, only that set of variables are used in your program calculation and the results on console is based on that set of variables. When your program works perfectly based on that set of variables, comment this set and uncomment another set of variables. Repeat this until you test your program with all 5 sets of variables.
 10. **IMPORTANT:** Every time that you run your program to test, your program must take the initial values for variables *assignments*, *midtermExam*, *quiz1*, *quiz2*, *finalExam*. Your program must calculate and show the **Total Weighted Average** and **WAT-on-Exams** **automatically** based on the values of those variables. Do not calculate them manually yourself and use Write() method to display on console the numbers you have calculated. This assignment is to exercise writing a program that does **calculation** and not just using Write() and WriteLine() to display some strings!
 11. Make sure your code looks neat and tidy. Press Ctrl+KD in Visual Studio to format your code based on coding styles.
 12. Pay attention that the results of calculations have only 2 decimal points, that is, 83.70 or 42.50 and not 83.71234 or 42.54567.
 13. Respect Naming Conventions for constants (ALL_CAPITALS) and variables (camelCase).

I have included the executable file of my program in the assignment zip file. It helps you to better understand what you must produce. This executable only displays the result of one of the above screenshots. Please note you cannot change the executable to display other screenshots as you do not have access to the source code of the program.

Checklist Before Submission

- All variables and consts are defined together on top of the methods (in this assignment, Main method)
- Consts have been used in the calculation of *weightedExams* and *totalWeightedAverage*
- Your program works correctly, and the results are identical to numbers in 5 screenshots provided to you

Submission

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

- A zip/archive the whole 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 would rename the zip file as a2_SaeedMirjalili.zip

Marking Guideline

Task	Points	Yours	Comments
The program works correctly with all 5 sets of values	25		
Proper variables defined and used properly	8		
Consts defined and used in calculations	7		
Correct totalWeightedAverage Expression is written	15		
Using Escape characters to display Banner properly	15		
Displaying the table properly using formatting rules	15		
Proper naming convention for variables(camelCase) and consts (ALL_CAPS)	15		
Total	100	0	

¹ a weighted average is different from a simple average.

For a Simple Average, you add up all the values, then divide by the total number of values.

Weighted Average is when values take different importance (weight), so you multiply by their weight (importance) then sum it all up, then divide by the total weight.

I explain it here: assume there are 3 values and 3 weights for them, the weighted average is calculated as

$$\text{weightedAverage} = (\text{value1} * \text{weight1} + \text{value2} * \text{weight2} + \text{value3} * \text{weight3}) / (\text{weight1} + \text{weight2} + \text{weight3});$$

whereas, a simple average is calculated as

$$\text{average} = (\text{value1} + \text{value2} + \text{value3}) / 3;$$

as you can see for simple average, no weight is considered. In fact, all values have the same weight.

for example, if values are (32,45,65) and weights are (20%, 35% and 45%) then

$$\text{weightedAverage} = (32 * 20 + 45 * 35 + 65 * 45) / (20 + 35 + 45) = (32 * 20 + 45 * 35 + 65 * 45) / (100) = 51.40;$$

or you can write equally your formula as:

$$\text{weightedAverage} = (32 * 0.20 + 45 * 0.35 + 65 * 0.45) / (0.20 + 0.35 + 0.45) = (32 * 0.20 + 45 * 0.35 + 65 * 0.45) / (1) = 51.40;$$

while the simple average of (32, 45, 65) is:

$$\text{average} = (32 + 45 + 65) / 3 = 142 / 3 = 47.33$$

as you can see simple average of (32, 45, 65) is 47.33 whereas weighted average of them with weights (20%, 35% and 45%) respectively is 51.45 and they are not the same.

Pay attention that in this assignment you have to calculate two weighted averages, one that includes all components of the course and one that excludes assignments from the calculation, so the second average is calculated out of 80% (not 100%) because the weight of assignments is 20% that is excluded, so the formula is:

$$\text{weightedExams} = (\text{midtermExam} * \text{MIDTERM_EXAM_PERCENTAGE} + (\text{quiz1} + \text{quiz2}) * \text{QUIZZES_PERCENTAGE} / 2 + \text{finalExam} * \text{FINAL_EXAM_PERCENTAGE}) / 0.8f;$$