

CS 161B: Programming and Problem Solving II

Assignment 1: Grade Calculations using Reference Parameters



Academic Integrity

You may NOT, under any circumstances, begin a programming assignment by looking for completed code on StackOverflow or Chegg or any such website, which you can claim as your own. Please check out the [Student Code of Conduct at PCC](#).

The only way to learn to code is to do it yourself. The assignments will be built from examples during the lectures, so ask for clarification during class if something seems confusing. If you start with code from another source and just change the variable names or other content to make it look original, you will receive a zero on the assignment.

I may ask you to explain your assignment verbally. If you cannot satisfactorily explain what your code does, and answer questions about why you wrote it in a particular way, then you should also expect a zero.

Introduction

Final Grade in this course is a combination of Assignments average, Midterm and Final Exam scores. They are weighted differently. Let's write a program to calculate the Final Calculated Score and the Letter Grade for a student in this course.

Grade Calculator

Grade Predictor • GPA Calculator • Eight Classes

History				89.85	B+	3.33
Assignment	% of Final Grade	Grade	Points			
Exam #1	20%	85	17			
Exam #2	20%	93	18.6			
Quizzes	15%	92	13.8			
Homework Assignments	10%	87	8.7			
Attendance	10%	100	10			
Final Exam	25%	87	21.75			

Homework Grades		Quizzes
72	90	
80	100	
96	85	

Cumulative Grade		94.75	A	3.88
	Grade	Letter Grade	GPA	Credits
Math	92	A-	3.67	3
Biology	96	A	4	3
Chemistry	91.8	A-	3.67	3
History	91.8	A-	3.67	3
English	98	A+	4	3
French	100	A+	4	3
Psychology	95	A	4	3
Band	93.4	A	4	3
				24

Purpose

In this assignment, you will be writing a program to calculate the final numerical score and the letter grade for a student in our class. You will read Assignment scores, and exam scores and calculate the final score and letter grade based on the weights of Assignments' scores and Exams' scores. The purpose of this assignment is to modularize your program.

After completing this assignment you will be able to:

- Write functions to modularize your program and reduce redundancy
- Write functions that take parameters by value and return values
- Write void functions that take nothing and return nothing
- Do data validation for bad input data for numbers
- Use the correct exit condition to exit out of a loop

Task

- ☐ Before you get started:
 - ☐ Check out Sample Assignment A01 - [Algorithmic Design document](#)
 - ☐ Check out Sample Assignment A01 - [SampleA01.cpp](#)
- ☐ Open the [Algorithmic Design Document](#), make a copy, and follow the steps to create your algorithm.
- ☐ You must express your algorithm as **pseudocode**.
- ☐ You will write a program to read the number of assignments from the user. Then read the assignment scores, the midterm and final exam scores. Calculate the average assignment scores and then calculate the final numeric score using the assignments average, the midterm and final exam scores. The assignments are weighted at 60%, the midterm and final exams are weighted at 20% each. The final score will be calculated out of 4 and the letter grade will be based on the following table. Check the Sample Run and do the calculations on paper and make sure you get the right answers before you write the code.

Letter Grade	4.0 Score	Level
A	$\geq 3.3 \leq 4.0$	Exceeds
B	$\geq 2.8 < 3.3$	Meets
C	$\geq 2.0 < 2.8$	Approaching
D	$\geq 1.2 < 2.0$	Not Yet
F	$0.0 < 1.2$	No Evidence

- ❑ Write functions to do the following. Each function should do ONLY what it is supposed to do, and no more unless specified. After writing each function, call the function in `main()` to test it before writing the next function.
 1. Welcome message (void function)
 2. A function called `int readInt(string prompt);` This function should be used any time you read any integers from the user. Use this function to read the number of Assignments from the user.
 3. A function called `void readScore(string prompt, double &num);` to read all the scores from the user. Any time you read a score you must call this function. The function must do data validation to make sure it is a numeric value, and that it is within the range of 0 to 4 inclusive. (Note: This function just reads a double and validates it - it can be used to read and validate any double number, the double number is sent by reference).
 4. A function called `double assignAverage(int numAssigns);` to read the Assignment scores from the user. This function must call `readScore` every time you read an Assignment score. You must use a `for loop` to do this. Then calculate and return the average score.
 5. A function called `void getInput(double &midtermScore, double &finalExamScore);` that calls the `readScore()` function and reads the 2 exam scores.
 6. A function called `double calcFinalScore(double assignAvg, double midtermScore, double finalScore);` that takes the Assignment average score, the midterm score and the final exam score, and calculates the final numerical score based on the appropriate weights for assignments and exams. The assignments are weighted at 60%, the midterm and final exams are weighted at 20% each.
 7. A function called `void calcLetterGrade(double finalScore, char &letter);` that takes the finalScore by value and a character for the letter grade by reference. Print the letter grade in `main()`.
 8. Your program must have function prototypes. Place the prototypes for your functions globally, after your `#includes`. **All functions must be implemented after `main()`.**
- ❑ You may not use a while true loop and break statements in the loop.
- ❑ Your program must check for valid input data. For example, it should not accept negative numbers for scores and all scores must be between 0 and 4 inclusive. Use loops to repeat until you get valid data from the user. Check out the [sampleA01.cpp](#) that has a `readInt` function that can be used to check for valid data. You can add the negative number check criteria to that function.
- ❑ Try not to have any redundant code (repeated code) in your program. That is the purpose of functions.
- ❑ All averages are displayed as with one digit precision.
- ❑ Use constants for all weights - 0.6, 0.2 etc for 60% and 20%

- ❑ **Do not use arrays or any vectors for this program. Use only the concepts and functions we have learned so far.**

Criteria for Success

- ❑ Look at the sample run below. Determine the function prototypes you will need for the various options (your list may include other options):
 - ❑ `welcome()` - void function, no parameters
 - ❑ You must have these functions: `readInt()`, `readScore()`, `assignAverage()`, `calcFinalScore()`, `calcLetterGrade()`, and `main()` with the appropriate parameters and return types mentioned above under [Task](#).
- ❑ Test your program using the following sample runs, making sure you get the same output when using the given inputs (in **blue**):

```
Welcome to my Final Grade Calculator!
Please enter the following information and I will calculate your
Final Numerical Grade and Letter Grade for you!
The number of assignments must be between 0 and 10.
All scores entered must be between 0 and 4.
```

```
Enter the number of assignments (0 to 10): 6
Enter score 1: 3.4
Enter score 2: 4
Enter score 3: 2.5
Enter score 4: 3.3
Enter score 5: 3.1
Enter score 6: 2.5
```

```
Enter your midterm exam score: 3.5
Enter your final exam score: 4
```

```
Your Final Numeric score is 3.4
Your Final Grade is A
```

```
Thank you for using my Grade Calculator!
```

```
Welcome to my Final Grade Calculator!
Please enter the following information and I will calculate your
Final Numerical Grade and Letter Grade for you!
The number of assignments must be between 0 and 10.
All scores entered must be between 0 and 4.
```

```
Enter the number of assignments (0 to 10): 3
Enter score 1: 3
Enter score 2: 4
Enter score 3: 2.5
```

Enter your midterm exam score: 2.5

Enter your final exam score: 2

Your Final Numeric score is 2.8

Your Final Grade is B

Thank you for using my Grade Calculator!

Welcome to my Final Grade Calculator!

Please enter the following information and I will calculate your Final Numerical Grade and Letter Grade for you!

The number of assignments must be between 0 and 10.

All scores entered must be between 0 and 4.

Enter the number of assignments (0 to 10): 12

Illegal Value! Please try again!!

Enter the number of assignments (0 to 10): 5

Enter score 1: 3.4

Enter score 2: 4

Enter score 3: 2.5

Enter score 4: 5.5

Illegal Score! Please try again!

Enter score 4: 3.5

Enter score 5: 3.1

Enter your midterm exam score: 3.5

Enter your final exam score: 4

Your Final Numeric score is 3.5

Your Final Grade is A

Thank you for using my Grade Calculator!

- ☐ Check out Sample Assignment A01 - [Algorithmic Design document](#)
- ☐ Check out Sample Assignment A01 - [sampleA01.cpp](#)
- ☐ Complete zyBooks **Chapter 9. CS161A: Functions pass by reference** activities.
- ☐ Complete all sections of your Algorithmic Design Document.
- ☐ Include **pseudocode** or a **flowchart** in part d of the design document.
- ☐ **Follow these Coding Construct Requirements:**
 - ☐ Must have all the functions mentioned above.
 - ☐ Your program must have function prototypes. Place the prototypes for your functions globally, after your #includes. **All functions must be implemented after main().**
 - ☐ Must do data validation for all numbers including scores.
 - ☐ Display all averages with one digit precision after the decimal point.

- ❑ Must have constants for Assignments and Exams scores weights (60% and 20%).
- ❑ Print a welcome and goodbye message.
- ❑ Please open and compare your work with the [grading rubric](#) before submitting.
- ❑ Remember to follow all [style guidelines](#).
- ❑ Download your Algorithmic Design Document as a PDF (File -> Download -> PDF), rename it to `a01.pdf`, and upload it to the D2L assignment by **Wednesday**.
- ❑ Upload your `a01.cpp` C++ source file to the D2L assignment by **Sunday**.
- ❑ Do your own work. Consult the syllabus for more information about academic integrity.

Additional Support

- ❑ Video: [Pass by Reference Tutorial](#)
- ❑ Check out Sample Assignment A01 - [Algorithmic Design document](#)
- ❑ Check out Sample Assignment A01 - [sampleA01.cpp](#)
- ❑ Post a question for the instructor in the Ask Questions! area of the Course Lobby.