

CS201 - Fall 2008 - Sabancı University

Homework #2 – Due October 30, 2008, Thursday, 13:00 (Sharp Deadline)

Please pay attention to the time of the deadline. It is not 19:00, but 13:00.

Introduction

The aim of this homework is to practice on parametric functions and if statements. The use of if statement is due to the nature of the problem (That is, you cannot finish this homework without using if statement). However, the use of functions is somehow enforced. That means it is possible to accomplish this homework without using functions, but it is a must to use them. The details about the use of functions in this homework are given later in this document.

In this homework, you will write a C++ program that calculates and displays total grades and success results (either "pass", "fail" or "make-up") of two courses after getting necessary inputs and input check. The total grade of a course is a weighted sum of midterm and final exams. The weights that you will use in this calculation are given in the next section. Your program will also decide whether you passed or failed the course or you will take a make-up exam. This decision will be given depending on your grades and the success conditions (rules) that are detailed in the next section. Your program should work for two courses; please see the subsequent sections for details.

Description

The program starts with displaying a welcome message that explains your program. The next phase is input entry for two courses. The inputs of each course are: *name of the course*, *midterm 1 score*, *midterm 2 score*, and *final exam score*.

First, a prompt is displayed to direct the user of the program to enter the name and exam scores of the first course. After this prompt, the user enters the necessary inputs for the first course. Then, a similar prompt is displayed for the second course and the user enters inputs of the second course.

After all the inputs are entered for both courses, your program first processes the data for the first course. After that, the second course is processed. In the rest of this section, the processing of a single course is explained. In your program, you have to repeat it for both courses.

For each course, your program should first check whether the inputs of that course are entered correctly or not (see the next section for details of this control). If there is a problem with the input entry of this course, then the program should display an error message and should not do anything else related with this course. If the inputs are entered correctly for a course, then your program should calculate the total grade for it. The total grade is the weighted sum of midterm and final exam scores of this course. The weight of Midterm 1 is 30%, the weight of Midterm 2 is 30% and the weight of Final exam is 40%. The next phase is to display some outputs about this course on the screen. As the output, first you have to display the total grade of the course which is calculated in the previous step. In this output, you have to refer to the course name that has been input beforehand. After that, you have to display the success result of this course. There are three options: either you "pass" the course, or you "fail" the course or you will be given a "make-up" exam. The corresponding rules are below.

- If the total grade of this course is at least 60, then you *PASS* the course.
- If the total grade of this course is between 60 and 45 (including 45, but excluding 60) and final exam score is at least 50, then you *PASS* the course.
- If the total grade of this course is between 60 and 45 (including 45, but excluding 60) and final exam score is below 50, then you will take *MAKE-UP EXAM*.
- If the total grade of this course is below 45, then you *FAIL* the course.

Please notice that, the abovementioned processing is explained for a single course. Your program should perform the same process for both courses. Moreover, as will be explained later in this document, this processing must be implemented as a set of parametric functions in your program.

Please see "Sample Runs" section in order to understand the flow of the program, inputs and the outputs in a better way.

Important Remarks about Inputs and Input Check

The inputs of the program are explained above. To repeat, these are *name of the course*, *midterm 1 score*, *midterm 2 score*, and *final exam score* (in this order). You have to have two sets of such variables and inputs; one for the first course, one for the second course. First you have to read the inputs of the first course, then for the second course. The course name is a string. The midterm and final exam scores can be real numbers. The order of inputs is explained above. It is extremely important to follow this order since we automatically process your programs and altered orders cause too much trouble. Please see "Sample Runs" section for some examples.

In your first homework, you were not expected to do any input checks, but in this homework you **have to** perform input check as detailed below.

The main rule is that all the numeric inputs (i.e. midterm and final exam scores) of this program must be nonnegative (i.e. greater than or equal to zero) and limited to hundred (smaller than or equal to hundred). The input checks of two courses must be done separately. For a course, if at least one of the exam scores is less than zero or greater than hundred, then your program should display an error message for this course. In the error message, you do not need to differentiate which input is entered wrong. A generic error message is sufficient. Please see "Sample Runs" section for some examples of wrong input and error message.

However, you are **NOT** expected to (and please **DO NOT** try to) re-input variables if erroneous inputs are entered. Such functionality requires loops or recursion that we have not covered yet.

All of the input operations (`cin` statements) will be in the main function (we have some technical reasons for doing so in this homework). Other operations like input check, calculations of total grades and evaluation of results will be in several user-defined functions. Moreover, you are going to pass some of the inputs and calculated values to these functions as parameters. These functions and parameter passing will be explained in the subsequent section.

Use of Functions (EXTREMELY IMPORTANT)

You have to follow the specifications below for function declaration and callings. The grading criteria will include proper use of these parametric functions. Do NOT use any global variables (variables defined outside the functions) to avoid parameter use. Unnecessary code duplication will cause grade reduction as well.

In the first homework you were not supposed to implement any functions. However, in this homework you are expected to (actually you have to) use some function(s). The guidance about using functions in this homework is below.

A total of four user-defined functions (other than main) must be implemented. On top of these functions, you may use other functions if you want.

Function 1: This function is going to display a prompt to describe the input information needed for a single course. You will not gather input with `cin` in this function. This function just prompts user about the information needed. User's input will be collected in the main function. Function 1 will be called twice (one for the first course; one for the second course) from the main function. As you can see from the sample runs given in the "Sample Runs" section, the only difference between these two calls are the words "first" and "second". This function should be a parametric one to cover this difference.

Hint: this function is going to have only one parameter of type string. The function will be called with argument "first" for the first course and with argument "second" for the second course.

Function 2: This function is for checking the validity of the exam scores for a single course. The exam scores of the course (scores of one course only, not of both courses) will be parameters. This function will check whether these scores are valid or not. If one or more scores are not valid, an error message will be displayed as mentioned in the previous section. If the exam scores are valid, Function 2 will just call Function 3 and the calculations will be made in Function 3. Thus, do not have any calculations in Function 2. This function (Function 2) will be called from main function twice (one for the first course; one for the second course).

Here remark that Function 2 have to take the exam scores (i.e. Midterm 1 score, Midterm 2 score, Final exam score) of a single course as parameters to check their validity and also to pass them to Function 3 for further processing. Moreover, Function 2 should also take the course name as parameter. This is needed due to two reasons: (1) to refer to the course name in the error message in case of invalid exam scores, and (2) to pass the course name to Function 3 as argument in case of valid exam scores.

Function 3: This function is the one where total grade calculation of a single course is performed. Thus, this function is going to be a parametric one, too. These parameters will be the name and the exam scores of a single course. Using these exam scores, you will calculate the total grade of the course using the weights given on the first page of this document. After calculating the total grade of the course, this function will not display anything, but call another function, which is Function 4, for that. Function 3 will be called from Function 2 as described above.

As you might easily see, the course name parameter of Function 3 is not directly used in calculation. You may think that this parameter is unnecessary, but this is not true. The course name parameter is needed to pass the course name information to Function 4 as argument.

Function 4: This function is going to display the necessary output for a single course. Function 4 will be called from Function 3 and will take three parameters: name of the course, total grade and the final exam score of a single course (midterm exam scores are not needed to be passed as parameters). In the output, you have to display the total grade of the course and the success result (pass/fail/make-up). Moreover, you have to mention the course name in the output as well. Please check out the sample runs below for examples. Since the total grade is passed to Function 4 as parameter, you must not recalculate it again in Function 4. In order to display the success result, you will need to apply the decision rules mentioned on the first page of this document. Please remark that not only the total grade, but also the final exam score is used in these rules. That is why Function 4 takes the final exam score as parameter.

Other than these four functions, you will also have the main function. The welcome message display, input operations (`cin` statements) will be in `main`. Moreover, you will call Functions 1 and 2 from `main`. Each of these functions will be called twice.

Needless to say that you have to name these four functions using meaningful identifiers, not as `function1`, `function2`, etc.

No abrupt program termination please!

Especially during the input check, you may want to stop the execution of the program at a specific place in the program. Although there are ways of doing this in C++, it is not a good programming practice to abruptly stop the execution in the middle of the program. Therefore, your program flow should continue until the end of the main function and finish there.

Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user. The introductory line (This program bla bla bla) is intentionally left incomplete. You are expected to use your imagination and creativity there to introduce your program. You may also change the prompts and output messages, but you have to display the required information for the results in the same order as here.

Sample Run 1

This program bla bla bla

Please enter the following data for your first course

Course name

Midterm 1 score

Midterm 2 score

Final score

CS201 86 82 60

Please enter the following data for your second course

Course name

Midterm 1 score

Midterm 2 score

Final score

Math201 43 71 52

Your grade for CS201 is 74.4

You passed CS201

Your grade for Math201 is 55

You passed Math201

Sample Run 2

This program bla bla bla

Please enter the following data for your first course

Course name

Midterm 1 score

Midterm 2 score

Final score

CS201 106 92 69

Please enter the following data for your second course

Course name

Midterm 1 score

Midterm 2 score

Final score

Math212 13 21 -2

Error! You entered invalid grade(s) for the course CS201

Error! You entered invalid grade(s) for the course Math212

Sample Run 3

This program bla bla bla

Please enter the following data for your first course

Course name

Midterm 1 score

Midterm 2 score

Final score

ENS204 100 92.5 89

Please enter the following data for your second course

Course name

Midterm 1 score

Midterm 2 score

Final score

CS202 73 81 122

Your grade for ENS204 is 93.35
You passed ENS204

Error! You entered invalid grade(s) for the course CS202

Sample Run 4

This program bla bla bla

Please enter the following data for your first course

Course name

Midterm 1 score

Midterm 2 score

Final score

SU101 -100 62 72

Please enter the following data for your second course

Course name

Midterm 1 score

Midterm 2 score

Final score

SU102 13.5 21 55.75

Error! You entered invalid grade(s) for the course SU101

Your grade for SU102 is 32.65

You failed SU102

Sample Run 5

This program bla bla bla

Please enter the following data for your first course

Course name

Midterm 1 score

Midterm 2 score

Final score

Kilpayi101 40 52 50

Please enter the following data for your second course

Course name

Midterm 1 score

Midterm 2 score

Final score

Kilpayi102 41 52 49.9

Your grade for Kilpayi101 is 47.6

You passed Kilpayi101

Your grade for Kilpayi102 is 47.86

You will take the make-up exam for Kilpayi102

Sample Run 6

This program bla bla bla

Please enter the following data for your first course

Course name

Midterm 1 score

Midterm 2 score

Final score

LIFE501 50 40 45

Please enter the following data for your second course
Course name
Midterm 1 score
Midterm 2 score
Final score
LIFE502 50.5 50 37.1

Your grade for LIFE501 is 45
You will take the make-up exam for LIFE501

Your grade for LIFE502 is 44.99
You failed LIFE502

Sample Run 7

This program bla bla bla

Please enter the following data for your first course
Course name
Midterm 1 score
Midterm 2 score
Final score
HR111 48 53 51

Please enter the following data for your second course
Course name
Midterm 1 score
Midterm 2 score
Final score
HR222 56 38.5 72.25

Your grade for HR111 is 50.7
You passed HR111

Your grade for HR222 is 57.25
You passed HR222

Sample Run 8

This program bla bla bla

Please enter the following data for your first course
Course name
Midterm 1 score
Midterm 2 score
Final score
Yat399 100 100 0

Please enter the following data for your second course
Course name
Midterm 1 score
Midterm 2 score
Final score
Uyan485 10 10 100

Your grade for Yat399 is 60
You passed Yat399

Your grade for Uyan485 is 46
You passed Uyan485

How to get help?

You may ask questions to TAs (Teaching Assistants) of CS201. Office hours of TAs are at the class website. All office hours will be held in MDBF L068 (assistant office hour room). Recitations will partially be dedicated to clarify the issues related to homework, so it is to your benefit to attend recitations.

What and where to submit (PLEASE READ, IMPORTANT)?

You should prepare (or at least test) your program using MS Visual C++ 6.0 (part of MS Visual Studio 6.0). We will use standard C++ compiler and libraries of the abovementioned platform while testing your homework.

It'd be a good idea to write your name and lastname in the program (as a comment line of course).

Submissions guidelines are below. Some parts of the grading process are automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

- ❑ Name your cpp file that contains your program as follows.

“SUCourseUserName_YourLastname_YourName_HWnumber.cpp”

Your SUCourse user name is actually your SUNet user name which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and lastname is Özbugsizkodyazaroglu, then the file name must be:

Cago_Ozbugsizkodyazaroglu_Caglayan_hw2.cpp

- ❑ Do not add any other character or phrase to the file name.
- ❑ Make sure that this file is the latest version of your homework program.
- ❑ Compress this cpp file using WINZIP or WINRAR programs. Please use "zip" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension.
- ❑ Check that your compressed file opens up correctly and it contains your cpp file. You will receive no credits if your compressed zip file does not expand or it does not contain the correct file.
- ❑ The naming convention of the zip file is the same as the cpp file (except the extension of the file of course). The name of the zip file should be as follows.

“SUCourseUserName_YourLastname_YourName_HWnumber.zip”

For example zubzipler_Zipleroglu_Zubeyir_hw2.zip is a valid name, but hw1_hoz_HasanOz.zip, HasanOzHoz.zip are NOT valid names.

- ❑ Submit via SUCourse ONLY! You will received no credits if you submit by other means (e-mail, paper, etc.).
 - 1) Click on "Assignments" at CS201 SUCourse (not the CS201 web site).
 - 2) Click Homework 2 in the assignments list.
 - 3) Click on "Add Attachments" button.
 - 4) Click on "Browse" button and select the zip file that you generated.
 - 5) Now, you have to see your zip file in the "Items to attach" list.
 - 6) Click on "Continue" button.
 - 7) Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.
- ❑ After submission, you will be able to take your homework back and resubmit. In order to resubmit, follow the following steps.
 - 1) Click on "Assignments" at CS201 SUCourse.
 - 2) Click Homework 2 in the assignments list.
 - 3) Click on "Re-submit" button.

- 4) Click on "Add/remove Attachments" button
- 5) Remove the existing zip file by clicking on "remove" link. This step is very important. If you do not delete the old zip file, we receive both files and the old one may be graded.
- 6) Click on "Browse" button and select the new zip file that you want to resubmit.
- 7) Now, you have to see your new zip file in the "Items to attach" list.
- 8) Click on "Continue" button.
- 9) Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

General Homework Rules

- ☐ Late penalty is 10% off of the full grade and only one late day is allowed.
- ☐ **Having a correct program is necessary, but not sufficient to get the full grade. Comments, indentation, meaningful and understandable identifier names, informative introduction and prompts, and especially proper use of required functions, unnecessarily long program (which is bad) and unnecessary code duplications (which is also bad) will also affect your grade.**
- ☐ Please submit your own work only (even if it is not working). It is really easy to find out "similar" programs!
- ☐ For detailed rules and course policy on plagiarism, please check out http://people.sabanciuniv.edu/levi/cs201/policy_plagiarism.html and keep in mind that

Plagiarism will not be tolerated!

Good Luck!