CS201 – Spring 2021-2022
Take-Home Exam 1
– CS201 Course Numeric Grade Calculation –
Due April 3rd, Sunday, 23:55 (Sharp Deadline)

Introduction

The aim of this take-home exam is to practice on the basics of C++ (input, output, assignment, arithmetic, etc.) and decision making (conditional if statements). The use of if statement is due to the nature of the problem; that is, you cannot finish this take-home exam without using decision making.

Description

In CS201, weights of the grading items on the course numeric grade are as given in Table 1. However, the weighted average is not the only criterion in letter grading, and thus, in the calculation of the course numeric grade.

Table1. Tentative Grading for CS201

GRADING ITEM	PERCENTAGE
Participation	7%
Take-Home Exams	25%
Midterm Exam	33%
Final Exam	35%

Here are some of the rules, which will be used in this take-home exam:

- If your exams' weighted average $(((MT * 0.33) + (FNL * 0.35))/0.68)$ is below 30, you will fail the course even if your total grade is equal or above the overall passing grade.

- All of the assigned take-home exams will be taken into consideration in the course numeric grade and each take-home exam will have equal weight.

Apart from the aforementioned rules, the contribution of the take-home exams to the course numeric grade is calculated according to the formula given below:

$$take\_home\_grade \begin{cases} take\_home\_averg & if \quad ratio \le 2 \\ take\_home\_averg \times (3 - ratio) & if \quad 2 < ratio < \\ 0 & if \quad ratio \ge 3 \end{cases}$$

$$ratio = submitted\_take\_home\_avr \ / \ weighted\_exam\_averg$$
$$weighted\_exam\_averg = (0.33 \times midterm\_grade + 0.35 \times final\_grade) \ / \ 0.68$$
$$course\_numeric\_grade = participation \times 0.07 + take\_home\_grade \times 0.25 +$$
$$midterm\_grade \times 0.33 + final\_exam\_grade \times 0.35$$

You may check the Syllabus or the course website to get the complete version of the grading schema of CS201 for the Spring 2021-2022 semester.

In this take-home exam, you will implement a C++ program that calculates the course numeric grade of a CS201 student. The inputs to your program will be (i) name of the student, (ii) midterm and final exam grades, (iii) take-home exam grades, and (iv) the number of days attended recitations of the student. Your program will then calculate the weighted exam average, take-home exam average, a ratio, percentage of participation and the course numeric grade, based on some conditions as described below.

During designing the algorithm for your program, you will follow the below listed *assumptions* in this take-home exam:

- There is no make-up exam for the regular midterm and final exams.

- There are a total of 4 (four) take-home exams, and the student has submitted a work for each of those take-home exams.

- There will be 13 weeks of recitation in a semester due to midterm week.

You may implement your algorithm all in the *main* function, i.e. you are not expected to write any user-defined functions, but of course you can write some, if you prefer to do so.

Your take-home exams will be automatically graded using GradeChecker, so it is very important to satisfy the exact same output given in the sample runs. You can

Inputs, Flow of the Program and Outputs

The inputs of the program and their order are explained below. <u>It is extremely important to follow this order with the same characters since we automatically process your programs</u>. ***Thus, your work will be graded as 0 unless the order is entirely correct***. Please see the "Sample Runs" section for some examples.

<u>The prompts of the input statements to be used has to be exactly the same as the prompts of the "Sample Runs"</u>.

At the beginning of your program, the user will be prompted to enter his/her name as the first input. The name of the user should be displayed whenever appropriate in the prompts and in the output text. Please see the "Sample Runs" section for some examples.

After the user enters his/her name, (s)he will be asked to provide his/her midterm and final exam grades. Once inputted, your program will calculate the weighted exam average as given below:

```
weighted_exam_grade = 0.33 x midterm_grade + 0.35 x final_grade
weighted_exam_avg = weighted_exam_grade / 0.68
```

If the weighted exam average of the user is below 30, (s)he will fail the course due to the course rules given before.

```
Hint :   avg_grade_to_pass = (30.0 * 0.33 + 30.0 * 0.35) / 0.68
```

In such a case, your program will display an appropriate message, and then, *your program will finish without even getting or prompting for the other inputs*.

On the other hand, if the weighted exam average of the user is equal to or greater than 30, then your program will continue. This time, the user will be asked to enter the grades of his/her 4 take-home exams. Once inputted, your program will calculate and display the average of those, and it will also calculate the ratio between weighted exam average and take-home exam average, which is to be calculated as given below:

$$ratio = take\_home\_avr / weighted\_exam\_avr$$

Thereafter, your program will calculate and display the points earned from the take-home exams depending on the schema below:

$$take\_home\_grade = \begin{cases} take\_home\_avr & if \quad ratio \leq 2 \\ take\_home\_avr \times (3 - ratio) & if \quad 2 < ratio < 3 \\ 0 & if \quad ratio \geq 3 \end{cases}$$

Then, your program will prompt for the recitation attendance of the user. You can calculate the percentage of participation as given below:

$$participation = recitation\_attendance / 13 * 100$$

Lastly, your program will calculate and display the course numeric grade of the user, which is to be calculated as given below:

```
course_numeric_grade =   participation * 0.07
                       + take-home_grade * 0.25
                       + midterm_grade * 0.33
                       + final_exam_grade * 0.35
```

You may assume that the user will always enter positive values correctly for all inputs. Therefore, you do <u>not</u> need to perform any kind of input checks.

Please refer to the "Sample Runs" section for some examples and further details.

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. You have to display the required information in the same order and with the same words and characters as below.

Sample Run 1

The purpose of this program is to calculate your CS201 grade. What is your name? ***Naz***
Hello Naz, what are your midterm and final exam grades? ***34 20***
Naz, your weighted exam average is less than 30, you will fail the course!

Sample Run 2

The purpose of this program is to calculate your CS201 grade. What is your name? *Ece*
Hello Ece, what are your midterm and final exam grades? *86 75.5*
What are the grades of your 4 take-home exams? *100 90.5 75 95*
You'll get all the points earned from take-home exams!
How many days have you attended recitations? *11*
Ece, your grade for CS201 is: 83.2593

Sample Run 3

The purpose of this program is to calculate your CS201 grade. What is your name? *Berk*
Hello Berk, what are your midterm and final exam grades? *50.5 35*
What are the grades of your 4 take-home exams? *96 88 90 75*
You'll get 82.7238 from the points earned from take-home exams!
How many days have you attended recitations? *10*
Berk, your grade for CS201 is: 54.9806

Sample Run 4

The purpose of this program is to calculate your CS201 grade. What is your name? *Can*
Hello Can, what are your midterm and final exam grades? *25 35*
What are the grades of your 4 take-home exams? *100 95 100 90*
You'll get NONE of the points earned from take-home exams!
How many days have you attended recitations? *9*
Can, your grade for CS201 is: 25.3462

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all take-home exams, unless otherwise noted.

- **How to get help?**

You can use GradeChecker (http://learnt.sabanciuniv.edu/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

You may ask questions to TAs (Teaching Assistants) or LAs (Learning Assistants) of CS201. Office hours of TAs/LAs are at the course website.

- **What and Where to Submit**

You should prepare (or at least test) your program using MS Visual Studio 2012 C++ (Windows users) or using xcode (macOS users).

It'd be a good idea to write your name and last name in the program (as a comment line of course). <u>Do not use any Turkish characters anywhere in your code (not even in comment parts).</u> If your name and last name is "İnanç Arın", and if you want to write it as comment; then you must type it as follows:
> // Inanc Arin

Submission guidelines are below. Since the grading process will be automatic, students are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be 0.

- Name your submission file as follows:

  - <u>Use only English alphabet letters, digits or underscore in the file names</u>. <u>Do not use blank, Turkish characters or any other special symbols or characters</u>.

  - Name your cpp file that contains your program as follows:
    "**SUCourseUsername_THEnumber.cpp**"

  - Your SUCourse user name is actually your SUNet username, which is used for checking sabanciuniv emails. Do <u>NOT</u> use any spaces, non-ASCII and Turkish characters in the file name (**use only lowercase letters**). For example, if your SUCourse username is

"**altop**", then the file name should be: **altop_the1**.*cpp* (please only use lowercase letters).

   ○ Do <u>not</u> add any other character or phrase to the file name.

● Please make sure that this file is the latest version of your take-home exam program.

● Submit your work **through SUCourse only**! You can use GradeChecker <u>only</u> to see if your program can produce the correct outputs both in the correct order and in the correct format. It will <u>not</u> be considered as the official submission. You <u>must</u> submit your work to SUCourse. You will receive no credits if you submit by any other means (email, paper, etc.).

● If you would like to resubmit your work, you should first remove the existing file(s). This step is very important. If you do not delete the old file(s), we will receive both files and the old one may be graded.

- **Grading, Review and Objections**

<u>Be careful about the automatic grading</u>: Your programs will be graded using an automated system. Therefore, you should follow the guidelines on the input and output order. Moreover, you should also use the same text as given in the "Sample Runs" section. Otherwise, the automated grading process will fail for your take-home exam, and you may get a zero, or in the best scenario, you will lose points.

<u>Grading</u>:

● There is NO late submission. You need to submit your take-home exam before the deadline. Please be careful that SUCourse time and your computer time <u>may</u> have 1-2 minutes differences. You need to take this time difference into consideration.

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

● If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.

● Please submit your **own** work <u>only</u>. It is really easy to find "similar" programs!

- Plagiarism will not be tolerated. Please check our plagiarism policy given in the [Syllabus](#) or on the [course website](#).

Plagiarism will not be tolerated!

<u>Grade announcements</u>: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

<u>Grade objections</u>: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your take-home exam from the email address provided in the comment section of your announced take-home exam grade or attend the specified objection hour in your grade announcement.
- Check the comment section in the take-home exam tab to see the problem with your take-home exam.
- Download the file you submitted to SUCourse and try to compile it.
- Check the test cases in the announcement and try them with your code.
- Compare your results with the given results in the announcement.

Good Luck!
Beyza Çandır & CS201 Instructors