CS201 – Fall 2021-2022
Take-Home Exam 1
– Classroom Restricted Capacity Calculator–
Due October 17ᵗʰ, 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-else statements). The use of if-else statements is due to the nature of the problem; that is, you cannot finish this take-home exam without using decision making.

Description

Fall 2021 semester is witnessing the return to campus and face-to-face lectures. However, the hazards and risks of Covid-19 still exist. Although lectures have returned to classrooms, restrictions have been applied to conform with social-distancing rules. Classroom capacities have been reduced to a percentage of their original capacity. In this take-home exam, you will implement a C++ program that performs calculations to determine whether all registered students can attend face-to-face lectures in the classroom. In the cases where not all registered students can attend the physical lecture, the program should also calculate how many students must be selected to attend on Zoom. The inputs to the program are below:

1. Name of classroom
2. Original room capacity
3. Coefficient (percentage) of allowed capacity
4. Number of students registered

You may implement your algorithm all in the *main* function, i.e. you are <u>not</u> 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 utilize GradeChecker (http://learnt.sabanciuniv.edu/GradeChecker/) to check whether your implementation is working in the expected way. To be able to use GradeChecker, you should upload all of your files used in the take-home exam (**only** *your_main_cpp* file for this take-home exam). Additionally, you should submit all of your files to SUCourse (**only** *your_main_cpp* file for this take-home exam) **<u>without zipping</u>** them. **Just a reminder, you will see a character ¶ which refers to a newline in your expected output.**

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 first prompted to enter the name of the classroom. The name of the classroom should be displayed in the final output after doing the necessary calculation. Please see the "Sample Runs" section for some examples.

Next, the user will be asked to provide the original room capacity. This number indicates how many students this classroom can fit before the applied restrictions. You should do an input check on the original room capacity such that if the user enters a negative number, the program should display an error message saying "You have entered an invalid value for room capacity." and then end the program.

Then, the program should take the allowed capacity coefficient input. This input represents the percentage of the original capacity that the classroom can fit. This value will be used to calculate the maximum number of students that can be allowed into the classroom using the following formula:

$$allowed\_capacity = original\_capacity * coefficient$$

Coefficient value needs to represent a percentage in the following range: 0% < percentage ≤ 100%. You should do an input check to make sure the value falls into that range otherwise, the program should display an error message saying "You have entered an invalid value for capacity coefficient." and then end the program.

If the coefficient input matches the criteria above the program will then take the final input: number of students registered. You should do an input check on the number of students registered such that if the user enters a negative number, the program should display an error message saying "You have entered an invalid value for registered students." and then end the program. This

number will be used to determine whether all registered students can fit by comparing the value with the allowed capacity. If all registered students can attend the physical lecture, the program should print an appropriate message. Otherwise, the program should calculate how many registered students should be selected to attend the zoom lecture since the classroom has reached its maximum capacity (after applying capacity reduction). This number can be calculated simply by the following formula:

```
zoom_students = registered_students - allowed_capacity
```

Then, it should print the amount along with an appropriate message.

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

IMPORTANT!

If your code does not compile, then you will get **zero**. Please be careful about this and double check your code before submission.

VERY IMPORTANT!

Your programs will be compiled, executed and evaluated automatically; therefore you should definitely follow the rules for prompts, inputs and outputs. See **Sample Runs** section for some examples.

● **Order of inputs and outputs** must be in the mentioned format.

Following these rules is crucial for grading, otherwise our software will not be able to process your outputs and you will lose some points in the best scenario.

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

Welcome to CS201 physical classroom capacity checker.
Please enter the room name: *G077*
Please enter the room capacity: *360*
Please enter the allowed capacity coefficient: *0.5*
Please enter how many students are registered to your course: *170*
G077 can allow all students to attend physical lectures.

Sample Run 2

Welcome to CS201 physical classroom capacity checker.
Please enter the room name: **G077**
Please enter the room capacity: **360**
Please enter the allowed capacity coefficient: **0.5**
Please enter how many students are registered to your course: **190**
G077 cannot allow all students to attend physical lectures.
For each lecture 10 students must be selected to watch on Zoom.

Sample Run 3

Welcome to CS201 physical classroom capacity checker.
Please enter the room name: **G015**
Please enter the room capacity: **-10**
You have entered an invalid value for room capacity.

Sample Run 4

Welcome to CS201 physical classroom capacity checker.
Please enter the room name: **G017**
Please enter the room capacity: **100**
Please enter the allowed capacity coefficient: **5**
You have entered an invalid value for capacity coefficient.

Sample Run 5

Welcome to CS201 physical classroom capacity checker.
Please enter the room name: **G077**
Please enter the room capacity: **100**
Please enter the allowed capacity coefficient: **0.8**
Please enter how many students are registered to your course: **-500**
You have entered an invalid value for registered students.

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:
                        // Baris Altop

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!
Gülşen Demiröz & Barış Altop