CS201 – SPRING 2020-2021
Take-Home Exam 4
– Boat Reservation Database –
Due 19th May, Wednesday, 23:55 (Sharp Deadline)

Introduction

The aim of this take-home exam is to practice on vectors, structs and file streams.

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 (**all** .cpp, .h and .txt files for this take-home exam). Additionally, you should submit all your .cpp and .h files to SUCourse **without zipping** them. **Just a reminder, you will see a character ¶ which refers to a newline in your expected output.**

The name of your main source (cpp) file should be in the expected format: "SUCourseUsername_THEnumber.cpp" (all lowercase letters). Please check the submission procedures of the take-home exam, which are listed at the end of this document.

To get help using GradeChecker you may ask questions to the list of your grader TAs: cs201gchelp@lists.sabanciuniv.edu.

Description

Your program will first start by reading a filename from the console, which will be the reserves file for boat reservations. The other two filenames "Boats.txt" and "Sailors.txt" must be hardcoded (i.e. you must define them in your program). If any of the files cannot be opened, just display the message "`Cannot open the files. Exiting…`" and exit from the program.

The columns in these files are separated by tab characters, and rows are separated by newline characters. First row of every file is the column identifier. You will read all of the three files, and load them into vectors. For the ease of the sorting part, we suggest that you read the reserves file "Reserves.txt" first and construct a single vector of reservations, and read the other two files on top of it to fill the missing fields (sailor name, boat name etc) which aren't given in the reservation information.

VERY IMPORTANT!

Your programs will be compiled, executed and evaluated automatically; therefore should definitely follow the rules for prompts, inputs and outputs. See **Sample R** 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 abl process your outputs and you will lose some points in the best scenario.

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. You should follow the input order in these examples and the prompts that your program will display **must** be **exactly the same** as given in the following examples.

Sample Run 1

```
Enter filename for reservation database: Reserves1.txt

2014-02-03 -> Zorba(16,10) has reserved CLIPPER(GREEN)

2014-02-18 -> Horatio(35,7) has reserved INTERLAKE(BLUE)

2014-03-12 -> Art(25.5,3) has reserved INTERLAKE(BLUE)

2014-03-12 -> Dustin(45,7.1) has reserved MARINE(RED)

2014-03-14 -> Brutus(33,1.1) has reserved INTERLAKE(RED)
```

Sample Run 2

```
Enter filename for reservation database: Reserves2.txt

2014-02-19 -> Dustin(45,7.1) has reserved INTERLAKE(BLUE)

2014-02-27 -> Rusty(35,10) has reserved TITANIC(BLACK)

2014-02-28 -> Lubber(55,8) has reserved INTERLAKE(RED)

2014-03-03 -> Brutus(33,1.1) has reserved MARINE(RED)

2014-03-10 -> Andy(25.5,8) has reserved CLIPPER(GREEN)
```

Sample Run 3

```
Enter filename for reservation database: Reserves.txt

Cannot open the files. Exiting...
```

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). Do not use any Turkish characters anywhere in your code (not even in comment parts). If your name and last name is "Barış Altop", 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:
    - Use only English alphabet letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
    - 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 NOT 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 not add any other character or phrase to the file name.

A sample line from the boat database is as follows:

```
101   INTERLAKE  BLUE
```

Here, 101 is the boat id, “Interlake” is the boat’s name and blue is the boat’s color.

A sample line from the sailor database is as follows:

```
22    Dustin   7.1 45.0
```

Here, 22 is the sailor id, “Dustin” is the sailor’s name, 7.1 is their rank and 45.0 is his age.

A sample line from the reservation database is as follows:

```
22    101   2014-02-19
```

Here, 22 is the sailor id of the sailor who reserved the boat, 101 is the boat id, and the date given is the date of the reservation.

Now pay attention to the id’s, sailor id of the reservation matches sailor Dustin’s sailor id, and boat id matches the boat named “Interlake”. So we want an output in the following format, displaying sailor info and boat info for the reservation.

```
2014-02-19 -> Dustin(45,7.1) has reserved INTERLAKE(BLUE)
```

For the overall output, you have to print all of the reservation information **sorted**. Your sorting algorithm must first take into account the dates of reservations (smaller date first), and if the dates are the same, then you should sort by the sailor’s names (whose name comes first in the alphabetical order first).

You need to use vectors, and you need to use structs.

The **structs** will be helpful to store the file contents within a single vector. You may assume each file you read is a separate entity.

Please do not try to process all three files at the same time. You should think of these files as database entries and construct a database containing all necessary information.

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 care
about this and double check your code before submission.

- 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 only to see if your program can produce the correct outputs both in the correct order and in the correct format. It will not be considered as the official submission. You must 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

Be careful about the automatic grading: 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.

Grading:

- 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 may 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 only. 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!

Grade announcements: 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.

Grade objections: 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!
Şevval Şimşek & Gülşen Demiröz & Barış Altop