CS201 – Fall 2022-2023
Homework 5
– CS201 Course – Movie Recommender (Class) –
Due January 6th, Friday, 23:55

Introduction

In this homework, you will modify Homework4 by adding a `Movie` class as opposed to using `struct`. You will also read the movie information from **a text file** instead of getting it from the user input.

Description

The program reads **N** (can be any number) movies from the file *(movies1.txt)*. Each line of the file contains the movie name, the year the movie was released and the genre of the movie, respectively, separated by spaces. **You cannot make any assumptions about how many words the movie name consists of, but you can assume that movie names will not contain numbers.** An example file is shown below:

```
The Lord of the Rings: The Fellowship of the Ring 2001 Fantasy
Amelie 2001 Romance
Spider-Man: Into the Spider-Verse 2018 Animation
Blue Valentine 2010 Romance
```

You should use a `class` to encapsulate the data of one movie; such as movie name, movie release year and movie genre. Then you should use a `vector` of movie objects with their data kept inside them.

After creating your vector, you should sort the vector according to the release year of movies *from newest to oldest*. If there are more than one movie released in the same year, you should sort these movies from A to Z ascending order according to the movie name.
DO NOT USE the algorithm library, instead implement a new free function using one of the sorting algorithms given in lecture slides.

The program that you are going to implement will operate based on the selection number that is input by the user. According to their selection, your program will either continue to recommend movies or exit the execution. There are 3 options:

1. `Select the genre of the movie that you want to watch`
2. `Select the year of the movie that you want to watch`
3. `Exit program`

If the user selects the 1st option, the user is prompted to enter a genre. After the user enters the genre, the movies belonging to that genre are displayed from newest to oldest. If there are movies released in the same year, they are displayed based on their names A to Z. You can assume that the genre input will be case-insensitive.

If the user selects the 2nd option, the user is prompted to enter a year or a range of years. When the user enters just a year, the program displays movies that are released that year, according to their names in ascending order. If the user enters a range of years, then the program displays the movies released in these years, from newest to oldest ones. Also if there are multiple movies released in the same year, they should be displayed based on the alphabetical order of their names, from A - Z.

For the 1st and 2nd options, if there is no matching movie, an appropriate message is printed.

Eventually, if the user selects the 3rd option, the program ends.

Input Check

1. You have to check the input for the option selection that must be in the range 1-3 (both included) and you can assume that it will always be given as an integer.
2. The name of the input file is also input from the *keyboard*. If the user enters a wrong file name that does not exist, your program should display an appropriate error message (check samples for error message) and ask the user again until the user enters a correct file name.
3. You can assume all the other inputs (genre, year, year range) are in the correct format.

IMPORTANT!

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

You should submit all of your files to SUCourse **without zipping** them.

VERY IMPORTANT!

Your homework will be automatically graded using **GradeChecker**, so it is very important to satisfy the exact same output given in the sample runs. You use (http://learnt.sabanciuniv.edu/GradeChecker/) GradeChecker to check whether your implementation is working in the expected way. To be able to use GradeChecker, you should upload all the files in your homework project. **Just a reminder, you will see a character ¶ which refers to a newline in your expected output.**

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

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.

For this homework, your submissions will also be processed manually to check whether the expected class functions are implemented. In case of missing member function implementations, you will lose points based on the severity of the deficiency. This manual processing does not mean that you may skip the format-related issues; the grade for the output of your program will still be obtained from the automated tool that we use for grading purposes.

It is a must to use the given movie class, any workaround will end up with a grade of zero (0) automatically.

You may use your codes from HW4, under the strict condition that you implement the movie class properly. As explained above, all submissions will be checked for proper implementation of the required class.

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.

See the General Guidelines below, about homework submissions and how to get help etc.

Movie Class

We will share the header file ***movie.h*** for the class you will write. You must write all seven (7) of the member functions in the class in this header file, not one more or one less. Below you may find the details of the member functions:

Constructors:

1. **Default Constructor:** In this default constructor you can assign an initial value to all private data members of the class.
2. **Parameterized Constructor:** In this parameterized constructor you must take all private data member values as parameters and assign them to the members.

Member Functions:

1. **setName:** A member function to set the name of a movie. This function will get a string as its only parameter and set the **name** *private data member* as the value of the parameter.
2. **getName:** Returns the name. (Accessor)
3. **setGenre:** A member function to set the genre of a movie. This function will get a string as its only parameter and set the **genre** *private data member* as the value of the parameter.
4. **getGenre:** Returns the genre. (Accessor)
5. **setYear:** A member function to set the year of a movie. This function will get an integer as its only parameter and set the **year** *private data member* as the value of the parameter.
6. **getYear:** Returns the year. (Accessor)
7. **print:** A member function that prints the name, year and genre of the movie, respectively, separated by spaces.

You **must** add all of the member functions listed above to the given movie class. You can also add free functions if you need them. For example, you might need another function to sort the vector.

Sample Runs

Below, we provide some sample runs of the program 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 the movie recommender program!

Please enter the movie list filename: movie.txt
Please check filename! Enter a correct movie list filename: movies1
Please check filename! Enter a correct movie list filename: movies1.txt
Please select your action:
1. Select the genre of the movie that you want to watch
2. Select the year of the movie that you want to watch
3. Exit program

Enter your choice: 1
Please enter the genre of the movie you want to watch: Fantasy

Fantasy movies from newest to oldest:
Movie name: The Chronicles of Narnia Release Year: 2005
Movie name: The Lord of the Rings: The Return of the King Release Year: 2003
Movie name: The Lord of the Rings: The Two Towers Release Year: 2002
Movie name: Harry Potter Release Year: 2001
Movie name: The Lord of the Rings: The Fellowship of the Ring Release Year: 2001
Movie name: The Green Mile Release Year: 1999
Movie name: Monty Python and the Holy Grail Release Year: 1975

Enter your choice: 1
Please enter the genre of the movie you want to watch: drama

There are no drama movies!

Enter your choice: 1
Please enter the genre of the movie you want to watch: HORROR

HORROR movies from newest to oldest:
Movie name: The Others Release Year: 2001

Enter your choice: 3
Thank you…
```

Sample Run 2

```
Welcome to the movie recommender program!

Please enter the movie list filename: movies1.txt
Please select your action:
1. Select the genre of the movie that you want to watch
2. Select the year of the movie that you want to watch
3. Exit program

Enter your choice: 2
Please enter the year of the movie you want to watch: 2001
Movies released in 2001 from A to Z:
Movie name: Amelie Genre: Romance
Movie name: Harry Potter Genre: Fantasy
Movie name: Ocean's Eleven Genre: Crime
Movie name: Shrek Genre: Animation
Movie name: Spirited Away Genre: Animation
Movie name: The Lord of the Rings: The Fellowship of the Ring Genre: Fantasy
Movie name: The Others Genre: Horror

Enter your choice: 2
Please enter the year of the movie you want to watch: 1940-1960
Movies released between the years 1940-1960 from A to Z with decreasing year ordering:
Movie name: It's a Wonderful Life Release Year: 1946 Genre: Romance

Enter your choice: 2
Please enter the year of the movie you want to watch: 1995-2005
Movies released between the years 1995-2005 from A to Z with decreasing year ordering:
Movie name: Pride & Prejudice Release Year: 2005 Genre: Romance
Movie name: The Chronicles of Narnia Release Year: 2005 Genre: Fantasy
Movie name: Eternal Sunshine of the Spotless Mind Release Year: 2004 Genre: Romance
Movie name: The Lord of the Rings: The Return of the King Release Year: 2003 Genre: Fantasy
Movie name: The Lord of the Rings: The Two Towers Release Year: 2002 Genre: Fantasy
Movie name: Amelie Release Year: 2001 Genre: Romance
```

```
Movie name: Harry Potter Release Year: 2001 Genre: Fantasy
Movie name: Ocean's Eleven Release Year: 2001 Genre: Crime
Movie name: Shrek Release Year: 2001 Genre: Animation
Movie name: Spirited Away Release Year: 2001 Genre: Animation
Movie name: The Lord of the Rings: The Fellowship of the Ring Release
Year: 2001 Genre: Fantasy
Movie name: The Others Release Year: 2001 Genre: Horror
Movie name: The Green Mile Release Year: 1999 Genre: Fantasy
Movie name: As Good as It Gets Release Year: 1997 Genre: Romance
Movie name: Good Will Hunting Release Year: 1997 Genre: Romance
Movie name: Life Is Beautiful Release Year: 1997 Genre: Romance
Movie name: Before Sunrise Release Year: 1995 Genre: Romance
Movie name: Se7en Release Year: 1995 Genre: Crime
Movie name: Toy Story Release Year: 1995 Genre: Animation

Enter your choice: 2
Please enter the year of the movie you want to watch: 1908
There are no movies released in 1908!

Enter your choice: 3
Thank you…
```

Sample Run 3

```
Welcome to the movie recommender program!

Please enter the movie list filename: movies1.txt
Please select your action:
1. Select the genre of the movie that you want to watch
2. Select the year of the movie that you want to watch
3. Exit program

Enter your choice: 4
Invalid action!

Enter your choice: 0
Invalid action!

Enter your choice: 3
Thank you…
```

Sample Run 4

```
Welcome to the movie recommender program!

Please enter the movie list filename: movies1.txt
Please select your action:
1. Select the genre of the movie that you want to watch
2. Select the year of the movie that you want to watch
3. Exit program

Enter your choice: 3
Thank you…
```

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all homework 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 can be found at SUCourse.

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).

You should also write your name and last name inside 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 "Gülşen Demiröz", and if you want to write it as comment; then you must type it as follows:

// Gulsen Demiroz

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, dot (‘.’) 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_hwnumber.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**) except dot ('.'). For example, if your SUCourse username is "**gulsend**", then the file name should be: **gulsend_hw5*.cpp*** (please only use lowercase letters).
    - Do not add any other character or phrase to the file name.
- Please make sure that this file is the latest version of your homework 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 want to resubmit your work, you should first remove the existing file(s). This step is very important as if you don't delete the old files, we 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 homework, 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 homework before the deadline. Please be careful that SUCourse time and your computer time may have 1-2 minute differences. You need to take this time difference into consideration.
- 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.
- 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.

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 homework from the email address provided in the comment section of your announced homework grade or attend the specified objection hour in your grade announcement.

- Check the comment section in the homework tab to see the problem with your homework.
- 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!
E. Beyza Çandır & Ekin Marlalı & CS201 Instructors