Due before:	11:59 PM (before Midnight) on Fri/Sat Sept 19/20 2025

Introduction to Systems Programming (Systems I)

Homework #3: C++ Movie DB

Due: Fri/Sat September 19/20 2025 before 11:59 PM

Email-based help Cutoff: 5:00 PM on Thu/Fri Sept 18/19 2025

Maximum Points: 25

Submission Instructions

This homework assignment must be turned-in electronically via Canvas. Ensure your C++ source code is named `Movie.h`, `Movie.cpp`, and *`homework3`*`.cpp`. Ensure your program compiles (without any warnings or style errors) successfully. Ensure you have tested operations of your program as indicated. Once you have tested your implementation, upload the above 3 source files onto Canvas via the CODE plug-in.

- The 3 C++ source files developed for this part of the homework.

Objective

The objective of this homework is to:

- Develop a custom C++ class to encapsulate `Movie` information.
- Continue to review the use of file streams for text file I/O
- Further understand the use of `std::unordered_map`.
- Continue to gain familiarity with developing C++ program involving simple string manipulation

Grading Rubric:

The program submitted for this homework **must pass necessary base case test(s) in order to qualify for earning any score at all**. Programs that do not meet base case requirements will be assigned zero score! Program that do not compile, **have even 1 method longer than 25 lines**, or just some skeleton code will be assigned zero score.

- **-1 Points**: for each warning generated by the compiler.
- **NOTE:** Violating CSE programming style guidelines is a compiler error! Your program should not have any style violations reported in CODE when you submit your solution.
- **Base case:** 13 points
- **Additional test:** 8 points
- **Formatting & documentation:** 4 points – Earned in full, only if the base **and additional functionality** are implemented.

Due before:	11:59 PM (before Midnight) on Fri/Sat Sept 19/20 2025

Background

In this homework you will be developing a simple program to process movie entries stored in a text file. Your program should enable the user to search the database and print relevant movie information. See sample outputs (at the end of this PDF) for commands and options.

Data file format

Note: The first line of the file is a comment-line. Read and ignore this line prior to processing entries from this file.

The supplied `movies_db.txt` has a space-separated data in the following columnar format:

```
moveID "Title" year "Genres" ImdbID Rating #Raters
```

Col Num	Name	Datatype	Description
1	`moveID`	`int`	Unique ID for a movie
2	`title`	`std::string`	A double-quoted title for the movie. Use `std::quoted` to read and write
3	`year`	`int`	The year when the movie was released
4	`genres`	`std::string`	A double-quoted list of genres for the movie. Use `std::quoted` to read and write
5	`imdbId`	`int`	The IMDB identifier for the movies
6	`rating`	`float`	An average movie rating (assigned by many reviewers)
7	`numRaters`	`int`	Number of reviewers who contributed to rating.

Starter Code:

For this homework no starter code is supplied. Try to use example code snippets from labs -- **specifically exercise #6** and lectures. However, an input text file is supplied. You need to copy the file(s) to your project folder for testing.

Note how each method is documented in the header file. You should follow the same style of Doxygen style documentation for your methods in `Movie.h` and methods in `main.cpp`.

If you find any aspect of documentation confusing or unclear, ensure you discuss it with your instructor.

Due before:	11:59 PM (before Midnight) on Fri/Sat Sept 19/20 2025

Homework Project Requirements:

The program for this project absolutely must be organized into the following 3 C++ files:

File name	Description
`Movie.h`	The header file to define a class called `Movie` with suitable instance variables to encapsulate the 7 fields for each `Movie` as listed in Data file format. This file must have: [**6 points**] 1. Constructor (with default values for all 7 fields) 2. Destructor (empty body) 3. Stream-insertion operator 4. Stream-extraction operator 5. A `to_string` method 6. An "`int getID() const;`" getter method
`Movie.cpp`	The source file that implements the methods associated with the `Movie` class. (yes, just like we did in exercise #6)
`homework3.cpp`	This class must contain all the necessary logic associated with loading, finding, searching movies as discussed below. This file must use an `std::unordered_map` to manage movie entries. (yes, similar to how it was done in the `main.cpp` in Exercise #6)

Functionality

The program for this homework must have the following functionality:

1. The program should load data from a fixed `"./movies_db.txt"` (exactly that path) file into an `unordered_map<int, Movie>`. [**3 points**]
2. The program should repeatedly obtain commands from the user (via `std::cin`) and perform the following operations [**3 points**]:

Cmd	Example	Description
`find <id>`	`find 1`	**Part of base case:** Prints information about the movie with given `id`. If `id` was not found it prints a suitable message (see sample outputs) [**3 points**]
`search "terms"`	`search "Iron Man"` `search "Horror|Mystery|Sci-Fi"`	Prints all movies with the given search term (*i.e.*, sub-string) if it occurs anywhere in the movie entry. The program should also print the count of matching movies. **See note below on how to read search terms** [**8 points**]
`exit`	`exit`	**Part of base case:** The program should stop operating. [**1 points**]

Note: You may assume the commands entered by the user will be in correct format. Hence, reading commands and **search terms** is simple using and `std::cin >> std::quoted(x)`, style of input. Do not overcomplicate input processing.

Due before:	11:59 PM (before Midnight) on Fri/Sat Sept 19/20 2025

Sample outputs

User inputs are shown in **bold**

Base case #1:

```
$ ./homework3
Enter a command:
exit
```

Base case #2:

```
$ ./harveysd_homework3
Enter a command:
find 1
1 "Toy Story" 1995 "Adventure|Animation|Children|Comedy|Fantasy" 114709
3.92093 215
Enter a command:
find 11723
Movie with ID 11723 not found in database.
Enter a command:
find 22322
Movie with ID 22322 not found in database.
Enter a command:
find 2232
2232 "Cube" 1997 "Horror|Mystery|Sci-Fi|Thriller" 123755 3.42593 27
Enter a command:
find 2
2 "Jumanji" 1995 "Adventure|Children|Fantasy" 113497 3.43182 110
Enter a command:
find 5
5 "Father of the Bride Part II" 1995 "Comedy" 113041 3.07143 49
Enter a command:
find -3
Movie with ID -3 not found in database.
Enter a command:
exit
```

Full test (order of search results may vary and that is ok)

```
Enter a command:
find 223322
Movie with ID 223322 not found in database.
Enter a command:
search "Iron Man"
59315 "Iron Man" 2008 "Action|Adventure|Sci-Fi" 371746 3.82447 94
167296 "Iron Man" 1931 "Drama" 22002 0.5 1
142056    "Iron    Man    &    Hulk:    Heroes    United"    2013
"Action|Adventure|Animation" 3221698 3 1
102125 "Iron Man 3" 2013 "Action|Sci-Fi|Thriller|IMAX" 1300854 3.5625
32
77561 "Iron Man 2" 2010 "Action|Adventure|Sci-Fi|Thriller|IMAX" 1228705
3.51064 47
102007 "Invincible Iron Man, The" 2007 "Animation" 903135 3 1
Found 6 matche(s).
Enter a command:
search "2014 \"Drama|Horror"
```

```
174053 "Black Mirror: White Christmas" 2014 "Drama|Horror|Mystery|Sci-
Fi|Thriller" 3973198 4.75 4
112515 "Babadook, The" 2014 "Drama|Horror|Thriller" 2321549 4.5 3
130052 "Clown" 2014 "Drama|Horror" 1780798 3.5 1
Found 3 matche(s).
Enter a command:
find 1
1 "Toy Story" 1995 "Adventure|Animation|Children|Comedy|Fantasy" 114709
3.92093 215
Enter a command:
exit
```

Turn-in:

This homework assignment must be turned-in electronically via Canvas CODE plug-in. Ensure your C++ source code is named `Movie.h`, `Movie.cpp`, and *`homework3`*`.cpp`. Ensure your program compiles (without any warnings or style errors) successfully. Ensure you have tested operations of your program as indicated. Once you have tested your implementation, upload the following onto Canvas:

- The 3 C++ source files you developed as part of this homework.

Upload all the necessary C++ source files to onto Canvas via the CODE plug-in. Do not submit zip/7zip/tar/gzip files. Upload each file independently.