

# Introduction to Systems Programming (Systems I)

## Homework # 3

**Due: Tuesday June 16 2020 before 11:59 PM**

**Email-based help Cutoff: 5:00 PM on Monday, June 15 2020**

Maximum Points: 50

### Submission Instructions

This part of the homework assignment must be turned-in electronically via Canvas. Ensure you name this document `MUId_homework3.docx`, where `MUId` is your Miami University unique ID. Complete the method shown for each problem. For each method, you can develop and test them in NetBeans and just copy-paste your solutions into this document.

Once you have completed answering the questions save this document as a PDF file (don't just rename the document; that is not the correct way to save as PDF) and upload it to Canvas

**General Note:** Upload each file associated with homework (or lab exercises) individually to Canvas. Do not upload archive file formats such as zip/tar/gz/7zip/rar etc.

### 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
  - Gain familiarity with C++ split design (.h .cpp)

### 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 for commands and options.

### Data file format

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

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

Col#	Name	Data Type	Description
1	moveID	int	Unique ID for a movie
2	Title	std::string	A double-quoted title for the movie
3	Year	int	The year when the movie was released
4	Genres	std::string	A double-quoted list of genres for the movie
5	imdbID	int	The IMDB (Internet Movie Data Base) 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

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 have: 1. Constructor 2. Destructor 3. Stream-insertion operator 4. Stream-extraction operator 5. A <code>to_string</code> method
Movie.cpp	The source file that implements the methods associated with the Movie class.
Homework3.cpp	This class <i>must</i> contain all the necessary logic associated with loading, finding, searching <i>and listing</i> movies as discussed below. This file use an <code>std::unordered_map</code> to manage movie entries. (all the required functionalities have already been implemented, your task is just to implement the <code>list()</code> function)

#### Sample outputs

User inputs are shown in **bold**

Enter a command:  
**exit**

Enter a command:  
**find 1**  
1 "Toy Story" 1995 "Adventure|Animation|Children|Comedy|Fantasy" 114709  
3.92093 215

Enter a command:  
**find 36000**

Movie with ID 36000 not found in database.

Enter a command:

**search Titanic**

4864 "Titanica" 1992 "Documentary|IMAX" 105601 2.5 1  
3403 "Raise the Titanic" 1980 "Drama|Thriller" 81400 4 1  
3404 "Titanic" 1953 "Action|Drama" 46435 3.58333 6  
1721 "Titanic" 1997 "Drama|Romance" 120338 3.41429 140  
Found 4 matche(s).

Enter a command:

**list**

.  
.  
4061 "The Man in the Moon" 1991 "Drama|Romance" 102388 4 2  
39414 "Shopgirl" 2005 "Comedy|Drama|Romance" 338427 3 7  
851 "Basquiat" 1996 "Drama" 115632 2.83333 6  
848 "Spitfire Grill, The" 1996 "Drama" 117718 3.28571 7  
78746 "Best Worst Movie" 2009 "Documentary" 1144539 4 1  
4051 "Horrors of Spider Island (Ein Toter Hing im Netz)" 1960 "Horror|Sci-Fi" 56600 0.5 1  
81949 "Romantics, The" 2010 "Comedy|Drama|Romance" 1403988 3 1  
842 "Tales from the Crypt Presents: Bordello of Blood" 1996 "Comedy|Horror" 117826 2.54167 12  
840 "House Arrest" 1996 "Children|Comedy" 116571 3.375 4  
839 "Crow: City of Angels, The" 1996 "Action|Thriller" 115986 2.83333 15  
7266 "Lost Skeleton of Cadavra, The" 2002 "Comedy|Horror|Sci-Fi" 307109 4 1  
7263 "Miracle" 2004 "Drama" 349825 3.35 10  
72178 "Welcome to Dongmakgol" 2005 "Comedy|Drama|War" 475783 4.5 1  
4042 "Alamo, The" 1960 "Action|Drama|War|Western" 53580 3.375 4  
833 "High School High" 1996 "Comedy" 116531 2 6  
163056 "Shin Godzilla" 2016 "Action|Adventure|Fantasy|Sci-Fi" 4262980 4 1  
7260 "Latter Days" 2003 "Comedy|Drama|Romance" 345551 3.5 1  
829 "Joe's Apartment" 1996 "Comedy|Fantasy|Musical" 116707 2.66667 9  
313 "Swan Princess, The" 1994 "Animation|Children" 111333 3.33333 3  
52245 "Blades of Glory" 2007 "Comedy|Romance" 445934 3.08824 17  
7256 "Touching the Void" 2003 "Adventure|Documentary" 379557 4 8  
72171 "Black Dynamite" 2009 "Action|Comedy" 1190536 3.5 2  
824 "Kaspar Hauser" 1993 "Drama|Mystery" 110246 4 1  
7251 "Where the Day Takes You" 1992 "Drama" 105810 4 1  
.  
.

**Due before:** 11:59 PM (before Midnight) on Tuesday June 16, 2020

---

4221 "Necessary Roughness" 1991 "Comedy" 102517 2.625 4  
Found 9742 entries.

NB: Note that . . means so on and so forth, you don't have to cout << “. .”

Attach some of your test run sample output (one sample page will be fine) in this docx file.

## Submission

- No late assignments will be accepted!
- This work is to be done individually
- The submission file will be saved with the name **HW3yourMUID.pdf**
- The submission file will be saved with the name **HW3yourMUID.cpp**
- Assignment is due Tuesday June 16, 2020 before Midnight
- On or before the due time, drop the *electronic copy* of your work in the *canvas*

Don't forget to Turn in the files! HW3\_yourMUID.pdf & HW3\_yourMUID.cpp

Due before: 11:59 PM (before Midnight) on Tuesday June 16, 2020

HomeWork3 - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Team Tools Window Help

mechlewg@os1.csi.miamioh.edu Debug

Projects Files Classes Services

HomeWork3 [mechlewg@os1.csi.miamioh.edu]

- Header Files
  - Movie.h
- Resource Files
- Source Files
  - Homework3.cpp
  - Movie.cpp
- Test Files
- Important Files
  - movies\_db.txt

Lab0 [mechlewg@os1.csi.miamioh.edu]

- Header Files
- Resource Files
- Source Files
  - main.cpp
- Test Files
- Important Files

Lab1 [mechlewg@os1.csi.miamioh.edu]

- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

Lab2 [mechlewg@os1.csi.miamioh.edu]

- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char\*\* argv) - Navigator

- MovieMap
- find(const MovieMap& db, const int id)
- list(const MovieMap& db)
- load(const std::string& filePath)
- main(int argc, char\*\* argv)
- search(const MovieMap& db, const std::string& searchStr)

Source

```
93 }
94 }
95 // Print the matching counts.
96 std::cout << "Found " << count << " matches(s).\n";
97 }
98
99 void list(const MovieMap& db) {
100 // Search the full info of a person to see if data is in there.
101 // use the range-based for loop (as discussed in lecture slides)!
102 int count = 0;
103 for (const auto entry : db) {
104 // Get the string form of the person's information.
105
106 // Check to see if the info contains the search terms
107 std::cout << entry.second << std::endl;
108 }
109 // Print the matching counts.
110 std::cout << "Found " << count << " matches(s).\n";
111 }
112
113 /*
114 * A simple main method to load and list of movies into a given
115 */
116 int main(int argc, char** argv) {
117 // First load the person database from a given file.
118 MovieMap db = load("movies_db.txt");
119 std::string cmd; // The command to process
120 // Repeatedly process command, until user enters exit.
```

Find: entry Previous Next Select

main while (std::cout << "Enter a command:\n",

Notifications Output

HomeWork3 (Build, Run) - mechlewg@os1.csi.miamioh.edu:22 x HomeWork3 (Run) - mechlewg@os1.csi.miamioh.edu:22 x

```
26717 "Begotten" 1990 "Drama|Horror" 101420 0.5 1
4219 "Girls Just Want to Have Fun" 1985 "Comedy" 89208 3.33333 3
69134 "Antichrist" 2009 "Drama|Fantasy" 870984 3.8 5
130840 "Spring" 2015 "Horror|Romance|Sci-Fi" 3395184 4.5 1
1010 "Love Bug, The" 1969 "Children|Comedy" 64603 3.05556 9
56152 "Enchanted" 2007 "Adventure|Animation|Children|Comedy|Fantasy|Musical|Romance" 461770 3.67647 17
4220 "Longest Yard, The" 1974 "Comedy" 71771 3.4 5
104875 "History of Future Folk, The" 2012 "Adventure|Comedy|Musical|Sci-Fi" 2245195 4 2
1011 "Herbie Rides Again" 1974 "Children|Comedy|Fantasy|Romance" 71607 2.9375 8
7438 "Kill Bill: Vol. 2" 2004 "Action|Drama|Thriller" 378194 3.86818 110
4221 "Necessary Roughness" 1991 "Comedy" 102517 2.625 4
Found 0 matches(s).
Enter a command:
exit

RUN FINISHED; exit value 0; real time: 7s; user: 50ms; system: 140ms
```