COMP 1602 – Computing Programming II

Assignment 2

Date Due: Sunday February 17, 2019 @ 11:55 pm

Description

This assignment requires you to write a program to manage information on movies and actors using structures. There are three files, one containing actor's information, one containing information on movies and one containing information on the cast of a movie. The program should provide a menu to enable the user to perform various queries on the data.

Actor File: actors.txt

The *actors.txt* file contains information on a set of actors, one actor per line. The amount of actors is unknown beforehand. Each line of data consists of an actor id, the actor's first name, the actor's last name, gender (a single character), age and his/her place of birth (string). For example,

```
NGJPZPPCQT     Jason          Momoa          M      39             Hawaii
AJJP86EQSE     Amber          Heard          F      32             Texas
HY8E5ESMWS     Tom            Hiddleston     M      37             London
FWVYFFHVP3     Josh           Brolin         M      50             California
N34YWZEMB9     Will           Smith          M      50             Pennsylvania
A1EVAM02EL     Michael        Jordan         M      31             California
AM02EL8SFB     Sylvester      Stallone       M      72             New_York
END
```

A special symbol, "`END`", indicates the end of the data. There are at most 100 actors in the file.

Movie File, movies.txt

The *movies.txt* file contains information on a set of movies, one movie per line. The amount of movies is unknown beforehand. Each line of data in the file contains the movie code, the name of the movie, the genre of the movie, the release date of the movie (in the format dd/mm/yyyy), worldwide gross to date and star rating. For example,

```
6687515        Aquaman                Action       12/12/2018     1067152007        7.4
6682515        Black_Panther          Sci_Fi       16/02/2018     1347071259        7.4
6687211        Venom                  Sci_Fi       07/10/2018     213515506         6.8
6687338        Creed_II               Drama        30/10/2018     17543141920       7.6
6587400        The_Dark_Knight        Action       24/07/2008     1004558444        7.4
6683410        Avengers:Infinity_Wars Sci_Fi       29/04/2018     2048709917        8.5
6688000        Alita:Battle_Angel     Adventure    02/14/2019     0                 0.0
6688106        Pet_Sematary           Horror       05/04/2019     0                 0.0
6688095        Miss_Bala              Action       01/02/2019     6800125           4.9
0
```

A movie id of 0 terminates the data in the file. There are at most 100 movies in the file.

Cast File: cast.txt

The *cast.txt* file contains information on the cast of a movie. Each line of data in the file contains a movie id corresponding to the movie id in *movies.txt*, an actor id corresponding to the actor id in *actors.txt* and the role played by the actor in that movie. For example:

```
6687515        NGJPZPPCQT     Arthur
6687515        AJJP86EQSE     Mera
6687338        A1EVAM02EL     Adonis
6687338        AM02EL8SFB     Rocky
6683410        HY8E5ESMWS     Loki
6683410        FWVYFFHVP3     Thanos
0
```

The first line of data shows that Jason Momoa played the role of Arthur in Aquaman. The second line of data shows that Amber Heard played the role of Mera in Aquaman.

As shown in the example, a movie id may appear zero times, one time, or several times in the file. Similarly, an actor id may appear zero times, one time, or several times in the file.

A movie id of 0 terminates the data in the file.

Your program should read the data from *actors.txt, movies.txt* and *cast.txt* and store them in appropriate arrays of structs.

After reading the data, your program should present the user with a menu with the following options:

1. Movie Information: Prompt the user for a movie name and print to the screen all the details of the movie, that is, the movie id, movie name, genre, release date and worldwide gross, all appropriately labelled.

2. Movie Cast: Prompt the user for a movie name and print to the screen a list consisting of the actor(s) name and the role played by each actor(s) in the movie.

3. Actor Information: Prompt the user for the actor's name and print to the screen, all the details of the actor, that is, his/her actor id, first name, last name, gender, place of birth and age appropriately labelled.

4. Actor Filmography: Prompt the user for an actor's name and print to the screen a list consisting of the name of the movie (s) and roles(s) played in the movie of all the movies the actor has acted in if the actor exists.

   If the actor exists but has not acted in any of the listed movies, display "Actor has not acted in any movie." If the actor does not exist, display an appropriate message.

5. Latest Releases: When the user chooses this option, a list of the names of all movies released within the last month is displayed. To get the current date use the getCurrentDate() function provided in `COMP1602Assignment#2.cpp`.

6. Coming Soon: When the user chooses this option, a list consisting of the name(s) of the movies and the release date of all the movies that have not yet been released is displayed. If there are no movies, an appropriate message is displayed.

7. Top-Rated Movies: When the user chooses this option a list of the name(s) of all the top-rated movie(s) and the star rating of the movie(s) is displayed. A top-rated movie has a star rating of at least 8.0 and a worldwide gross of over 1 billion dollars.

8. Exit

The program should end only when option 8 is chosen. If the user enters an invalid option, an error message should be printed and the menu re-displayed.

Files to Download

`actors.txt:` actor file
`movies.txt:` movies file
`cast.txt:` cast file
`COMP1602Assignment#2.cpp:` Use this file as a starting point for your solution to this assignment. Write the code for the assignment in this `.cpp` file.

These four (4) files **must** be saved to a folder. The name of the folder is your student id number.

Note

Additional functions may be used.

Submission Instructions

- Upload the *zipped folder* containing ***only*** the **.cpp** files and .**txt** files to myElearning no later than **Sunday 17th February 2019 at 11:55pm**.
- Absolutely no late or emailed submissions will be accepted.