Data Structures & Program Design Final Project Final Report Document:

Academy Awards Database

By Julie Durbin

Due: 11 December, 2019

**Summary of Main Functions:**

* Read actors
  + Reads the actor file, stores the data into a temp actor, and pushes it into the actor vector
  + Uses getline and stores everything as a string
  + Has to use the comma as a delimiter
  + Splices the name of the actor into first and last name to ensure correct sort by last name using string stream and getline
    - If no last name (eg/ Cher) we save that as the last name so it will alphabetize correctly
  + Makes sure the first line starts with correct character so we’re not accidentally reading picture file into actor vector
  + Gets rid of /r at the end of the film if it’s there
* Read pictures -- read picture file
  + Reads the actor file, stores the data into a temp picture, and pushes it into the actor vector
  + Uses getline and stores everything as a string
  + Has to use the comma as a delimiter
  + Deletes beginning space in front of each picture name if it’s there
* Sort Actors
  + Sorts a given vector of actors by name, year, win, award or film
  + Uses struct in actors.h for third parameter of STL sort function
  + Picks type of sort by integer option for the field selected by the user
* Sort pictures -- sort pic vector (or 2ndary search vector) based on given field
  + Sorts a given vector of pictures by name, year, nominations, genre1, and rating
  + Uses struct in actors.h for third parameter of STL sort function
  + Picks type of sort by integer option for the field selected by the user
* Print Actors -- prints actor vector w headings
  + Prints a vector of actors based on the size
  + Uses a print actor function in actors.cpp that prints a single actor with correct spacing
  + Prints header with spacing above the vector of actors
* Print Pictures
  + Prints a vector of pictures based on the size
  + Uses a print picture function in pictures.cpp that prints a single actor with correct spacing
  + Prints header with spacing above the vector of actors
* Complete Search Actors -- searches for actor based on exact field
  + Returns a vector of results for secondary search
  + Takes in string to search and integer value to sort by field
  + Searches entire vector for exact match and then puts it into the results vector
* Complete Search Pictures --searched for picture based on exact field
  + Returns a vector of results for secondary search
  + Takes in string to search and integer value to sort by field
  + Searches entire vector for exact match and then puts it into the results vector
* Partial Search Actors
  + Takes in a string to search for, an integer value that represents the field to search, and a vector of actors, which will allow us to do a secondary search on the results vector
  + Uses string.find() function. If there is no find, it returns string::npos and we add nothing to the vector. If there is a match, add it to the results vector
* Partial Search Pictures -- searched for pic based on partial input for a given field
  + Takes in a string to search for, an integer value that represents the field to search, and a vector of pictures, which will allow us to do a secondary search on the results vector
  + Uses string.find() function. If there is no find, it returns string::npos and we add nothing to the vector. If there is a match, add it to the results vector
* Add actors
  + Prompts user to input actor information for each field
  + Protects against bad input with while statements
  + Best way to protect against bad integer/float input is taking it in as an int/float and checking it against cin.fail() then using the to\_string() function to convert it
  + Uses constructor to make a new actor and push into vector
  + Resorts the vector by name after inputting & prints
  + Asks to add more or return to main menu
* Add pictures
  + Prompts user to input picture information for each field
  + Protects against bad input with while statements
  + Best way to protect against bad integer/float input is taking it in as an int/float and checking it against cin.fail() then using the to\_string() function to convert it
  + Uses constructor to make a new picture and push into vector
  + Resorts the vector by name after inputting & prints
  + Asks to add more or return to main menu
* Delete actor
  + User must enter exact full name of actor to delete/modify
  + Prompts user to continue or search in case they need to check the name
  + Instructs the user how to enter the input correctly
  + Asks if they want to delete or modify
  + If delete, search vector for that entry and erases that entry
  + If no matches found, display error message
  + If modify, asks the user which field to modify
  + Searches for selected actor and calls modify actor function with field selection, then resorts
  + Asks to delete or modify another actor or display current actors and go back to main menu
* Modify actor
  + Takes in a field and an index of the actor to modify
  + Uses switch to modify based on the field
  + Makes sure user enters in relatively appropriate input with while statements
  + Makes modification with set member functions for actor
* Delete picture
  + User must enter exact full name of picture to delete/modify
  + Prompts user to continue or search in case they need to check the name
  + Instructs the user how to enter the input correctly
  + Asks if they want to delete or modify
  + If delete, search vector for that entry and erases that entry
  + If no matches found, display error message
  + If modify, asks the user which field to modify
  + Searches for selected picture and calls modify actor function with field selection, then resorts
  + Asks to delete or modify another picture or display current actors and go back to main menu
* Modify picture -- modifies a selected picture based on a given field
  + Takes in a field and an index of the picture to modify
  + Uses switch to modify based on the field
  + Makes sure user enters in relatively appropriate input with while statements
  + Makes modification with set member functions for picture
* Write Actor File
  + Opens a new file
  + Uses a for loop to write actors to the new file
  + Separates fields with commas for csv file and new line character after each actor
  + Goes back to main menu afterwards
* Write Picture File -- saves modified vector by writing to new file
  + Opens a new file
  + Uses a for loop to write picture to the new file
  + Separates fields with commas for csv file and new line character after each picture
  + Goes back to main menu afterwards

**Menu Functions & Flow:**

* Start Screen
  + Asks user to enter filepath for both databases
  + Calls read actor & read picture function and passes the filepath
  + Restarts program and calls itself if it failed to read either file
  + If successful, goes to main menu
* Main Menu
  + Asks to go to picture menu, actor menu, or quit
* Actor Menu
  + Options to sort, search, add, delete/modify, write file, or return to main menu
  + Calls appropriate function in switch case based on selection
* Sort Actor Menu
  + Select field to sort by
  + Option to go back to main menu
  + Either calls sort function and passes field selection or goes back to main menu
  + Goes back to sort function if user didn’t select to go back to main menu
* Search Actor Menu
  + Firsts asks for complete search or partial search
  + Select field to search by
  + Asks user for query
  + Asks to search again, run a secondary search based on the results, go to the delete/modify function, or the main menu
* Picture Menu
  + Options to sort, search, add, delete/modify, write file, or return to main menu
  + Calls appropriate function in switch case based on selection
* Sort Picture Menu
  + Select field to sort by
  + Option to go back to main menu
  + Either calls sort function and passes field selection or goes back to main menu
  + Goes back to sort function if user didn’t select to go back to main menu
* Search Picture Menu
  + Firsts asks for complete search or partial search
  + Select field to search by
  + Asks user for query
  + Asks to search again, run a secondary search based on the results, go to the delete/modify function, or the main menu

**Status of Program:**

* Everything should be working correctly
* All requirements should be met

**Status on CSE Grid:**

* After testing, the program appears to work well on CSE grid without any errors, unexpected terminations, ect
* There is one warning for a multiple character constant (\r) which I use to get rid of the \r at the end of the film string in the actor file if it’s there
* Occasionally the cin will have trouble and say there is invalid input when input was fine. This seems to clear up when you reenter the input. It’s probably just a problem with cin

**Improvements That I Could Have Made:**

* I wish that I had made an actorDatabase.cpp file (with appropriate header file) and a pictureDatabase.cpp file (with appropriate header file). This would have helped me organize my functions better and would have been more modular. Unfortauntely, I realized this less than 24 hours before the final project was due. I am disappointed in the size of Database.cpp and it is quite difficult to fully navigate the file and see all of the functions because most functions are crammed into that file.
* I could have implemented this by putting the start menu and the main menu and main and then could have gone from there, or even created a separate menu class. I thought about putting most of the functions in actors.cpp and pictures.cpp but that wouldn’t have worked so well because the class doesn’t have its own vector of classes like the database class has
* My code could have overall been cleaner and more concise. I think having most of the functions in database.cpp made it hard to traverse each function which also made it difficult to edit. For example, I probably wrote too much code in the search function, which asks for a partial search or complete search and then repeats the same chunk of code for each choice, rather than writing one chunk of code that then uses an if statement to either call the partial search function or the complete search function