

CS162 Project 2

Goals for This Project:

- Break tasks down into functions
- Master pass by reference or pass by value
- Use character arrays and `<cstring>`
- Use `struct` to model composite data type
- Use array of objects to model a collection
- File input/output

Problem You Need to Solve for This Lab:

You are asked to write an app to keep track of a relatively small music library. The app should load song information from a data file once the app is started. It should allow user to view, add, remove, and search for songs. The app should save the data back to the same data file when the program exits.

What Your Program Should Do:

Write an interactive text based menu interface (using a loop) that will allow the user to

- Enter information for a new song
- Display information for all the songs in the database with index for each song
- Remove a song by index
- Search for songs by a certain artist
- Search for songs by a certain album
- Quit

For each song, you need to keep track of:

- title
- artist
- duration
- album

Allow the program to keep looping until user wants to quit. When the program starts, it should load the tasks from external file ("**songs.txt**") into memory. When user enters information about the new song, the program needs to read them in, save them in memory and eventually write them to the external data file ("**songs.txt**"). The file format could look like:

Stereo Hearts;Gym Class Heroes;3;34;The Papercut Chronicles II
Counting Stars;OneRepublic;4;17;Native

The ';' is used as a delimiter or field separator. Each record ends with a new line character. Also the above sample data came from my teen son, not a reflection of your instructor's music taste ☺

Some Implementation Requirements:

1. Write at least **four** functions **WITH** arguments for this assignment.
2. Use struct named **Song** to model each song
3. Use array of structs to model the collection of songs.
4. Hint: In this assignment, some data fields may have multiple words in it. Therefore, you now **SHOULD** read using the 3 argument version of `get`.
5. Watch out. When using the 3 argument version of `get` you need to make sure to remove the delimiter or newline. Therefore, anytime you read (even a confirmation message), make sure to eat the newline using `cin.ignore(...)` !
6. Make sure to have a delimiter written between each item in the file – like a newline. This will be important when you read the information back from the file.
7. **For submission, your data file should contain a sufficient set of test data. It should have test cases for same artist with multiple songs and same album with multiple songs in it.**