

CSC 249 Data Structures and Algorithms

Project – 1

IMPORTANT:

- Your instructor and/or lab assistants have been available to assist with all hands-on activities. Students are expected to complete this project without assistance from others (this includes receiving assistance from individuals inside or outside of CPCC).
- You must include your name, date, and the purpose of the program in the beginning of your code.
- Must include short comments throughout your code (each section) to describe variables, constants, classes, etc.

Instructions

Use the following information to create a program that allows the user to manipulate a music playlist and view its content.

Your program should contain 3 separate classes: a **Song** class, a **Playlist** class, and a **Demo** class.

In this program, the Playlist class will function as a doubly linked list that contains Song objects. The Demo class will be used to make modifications to the Playlist.

Song Class

This class should contain the following fields:

- name: String
- artist: String
- album: String
- length: int
 - o This is the length of the song in seconds
- next: Song
 - o This is a reference to the song that follows this song in the list
- previous: Song
 - o This is a reference to the song that precedes this song in the list

The Song class should contain the appropriate getters and setters for these attributes, a no-arg constructor, and a constructor that accepts the following arguments: name, artist, album, and length.

Playlist Class

This class will act as our doubly linked list. This class should contain the following attributes:

- head: Song
 - o A dummy node (a blank song object that will represent the first song in the playlist)
- tail: Song
 - o A dummy node (a blank song object that will represent the last song in the playlist)

CSC 249 Data Structures and Algorithms

Project – 1

- size: int
 - The number of songs in the playlist

Note: Make sure to review the section in your book on Dummy Nodes - they should help simplify the logic required for methods that operate on the Playlist class.

The Playlist class should contain a no-arg constructor which initializes the size, head and tail attributes. This no-arg constructor should also properly connect the head and tail nodes of the doubly linked list.

The Playlist class should also contain the following methods:

- A method that accepts a Song object and adds it to the end of the Playlist.
- A method that accepts 2 Song objects (a new Song to be added, and a Song that is currently playing from the playlist). This method will insert the new Song into the playlist at the position directly after the Song that is currently playing.
- A method that accepts a Song object, returns nothing, and removes the Song from the playlist.
- A toString method that prints out the contents of the Playlist. This toString method should also calculate and display the number of songs in the playlist, as well as the length of the playlist in minutes and seconds.

Demo Class

This class will serve to accept user input and manipulate the Playlist doubly linked list.

The Demo class should instantiate a Playlist object and display a menu to the user that will allow for the following actions:

1. Add a Song to the Playlist
2. Insert a new Song after the Song that is currently playing
 - If a song is being added/inserted to an empty playlist, that song should then be set to be the one that is currently playing.
3. Print the contents of the Playlist
4. Display the current Song
 - When displaying the current song for the first time, the first song in the playlist should be displayed.
 - To accomplish this, create a Song reference variable and assign it the first Song in the Playlist
5. Remove the current song
 - When removing the current song, the new current song will be the song that directly follows the removed song.
6. Skip to the next song
 - If you are at the end of the Playlist, the next Song should be the first Song in the playlist.
7. Return to the previous song

CSC 249 Data Structures and Algorithms

Project – 1

- If you are at the beginning of the Playlist, the previous Song should be the last Song in the playlist.

8. Exit

Validate the user's input for the menu option. The user should be required to enter a valid menu option before advancing.

If any of the options chosen would not work due to there being no applicable songs in the playlist, an appropriate error message should be displayed.

The user should be allowed to repeatedly select menu options until they choose to Exit.