Assignment 1 — Music Tracks and Playlist Management System

Instructions

1- Students will form teams of **2** students **from the same lab group or across groups.**

2- If more than 2 students submit the assignment, all the team members will get zero for the assignment.

3- Deadline of submission is **November 14 th at 11:59 pm**.

4- Submission will be done through a form link that will be posted on the Google classroom.
5- No late submission is allowed.

6- Submission will be done through a Google form link. Such link will be posted on the Google classroom before the deadline. Once you submit through the form, you will receive a copy of your submission by email. It is your responsibility to make sure that your submission went through properly. If you found a problem in your submission, you can still edit it before the deadline.

7- No submission through e-mails.

8- You will develop the needed .cpp and .h files that should **all** include a block comment containing students' IDs and names. Those files should be put in a folder named **Assign1_firstStudentID_SecondStudentID** and compress them to a .zip file with the same folder name. The compressed file would be the file to be delivered. Failing to abide by this naming convention will result in grades' deduction

9- Students who attend the same lab slot can form a group like S7 and S8. In this case name the file with any of the 2 groups (S7 or S8 for ex.)

10- In case of Cheating you will get a negative grade whether you give the code to someone, take the code from someone/internet, or even send it to someone for any reason.

11- You have to write clean code and follow a good coding style including choosing meaningful variable names.

Task

The Music Tracks and Playlist Management System allows users to create, copy, compare, and manipulate playlists and the songs they contain.

1. Struct Song

Define a struct named Song with the following attributes:

- ☐ string title — title of the song
- ☐ double duration — duration of the song in minutes

2. Class MusicTrack

Design a class called MusicTrack to manage playlists dynamically.

Private Attributes:

1. Song* playlist — dynamic array of Song objects representing the current playlist.

2. int playlist_size — number of songs currently in the playlist.

3. Functions

1) createPlaylist() — default constructor that dynamically allocates memory for a playlist and initializes its size.

2) addNewSongs() — receives one or more new songs and adds them to the playlist (with resizing if needed).

3) removePlaylist() — Destructor that frees dynamic memory and confirms removal.

4) copyPlaylist() — copy constructor for deep copying playlists.

5) totalPlaylistsCreated() — static function to return total playlists created.

6) longestSongInAllPlaylists() — static function returning the longest song and its title among all playlists.

7) operator>= — friend function to compare two playlists (based on number of songs).

8) operator[] — play song at a specific index.

9) operator+ — returns a new playlist with common songs between two playlists.

10) operator- — returns a new playlist with unique songs from the first playlist.

11) operator-- — removes the last song from the playlist (postfix).

12) operator<< — friend function to display all songs in a playlist.

4. Menu Implementation

Implement a menu in the main program that allows the user to choose from the following options:

1. Create a new playlist

2. Add new songs to a playlist

3. Remove a playlist

4. Copy a playlist

5. Display total playlists created

6. Show the longest song among all playlists

7. Compare two playlists

8. Play a song by index

9. Display common songs

10. Display unique songs

11. Remove last song

12. Print all songs

13. Exit

6. Grading Rubric (Total 100 Marks)

• Operator overloading (>=, +, -, [], --, <<) — 60 marks

• Static members and functions — 10 marks

• Dynamic memory management (allocation, resizing, cleanup) — 10 marks

• Menu-driven functionality and demonstration — 10 marks

• Constructors and Destructor — 10 marks