CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

(UIC)

Project Report

ON

Playlist-Based Music Player Application in C++

Program Name: BCA

Subject Name/Code: Object Oriented Programming(24CAH-201)

Submitted by:

Name: Inderjeet Kaur

UID: 24BCA10390

Section: 24BCA 2

Group: B

Submitted to:

Name: Ms. Rashmi Saluja

Designation: Assistant Professor

Table of Contents

Acknowledgement

I would like to express my sincere gratitude to everyone who contributed to the successful completion of this project. I am deeply thankful to my instructor for providing guidance, support, and encouragement throughout the development of this work. Their valuable suggestions and insights helped me understand the concepts more clearly and implement them effectively.

I am also grateful to my institution for offering the necessary resources and a supportive learning environment that allowed me to explore and apply programming concepts practically. This project provided me with an opportunity to enhance my understanding of Object-Oriented Programming in C++ and its real-world applications.

Finally, I would like to thank my family and friends for their constant motivation and support during the course of this project. Their encouragement played an important role in helping me stay focused and determined.

This project has been a significant learning experience, and I am thankful to all who guided and supported me throughout the process.

Introduction

Technology plays a vital role in enhancing convenience and efficiency in daily activities. This project is a simple implementation of a Music Player Management System using C++ programming, focusing on core Object-Oriented Programming (OOP) concepts.

The system enables users to manage a personal music playlist persistently. Users can add new songs, view the full playlist, play a selected song, and delete songs from the playlist. The implementation emphasizes clarity, modularity, and user interaction, making it an effective example of applying theoretical programming knowledge to practical problem-solving.

Project Summary

This project focuses on the development of a simple and interactive Music Player system using the C++ programming language. The system models the basic mechanism for managing a personal music library, demonstrating how Object-Oriented Programming (OOP) principles can be used to structure a real-world application in a systematic way.

The program structure uses two main classes: the Song class and the MusicPlayer class. The system achieves data persistence by loading and saving the playlist to an external file ("playlist.txt"). This implementation demonstrates programming concepts such as classes and objects, stream manipulation, and file handling.

Objectives of the Project

The main objective of this project is to design a basic music management system that allows a user to maintain and interact with a persistent playlist. The additional objectives include:

1. To apply Object-Oriented Programming concepts to construct a modular program structure.
2. To demonstrate the use of user-defined classes (Song and MusicPlayer) with relevant data members and member functions.
3. To incorporate file handling for recording and retrieving playlist details in a persistent manner.
4. To create a user-friendly interface that interacts smoothly with the passenger.

Features of the System

The project contains several useful features:

1. Playlist Management: The system offers a menu with options for Add Song, Show Playlist, Play Song, Delete Song, and Exit.
2. Song Object Definition: Each song is encapsulated in a Song class holding its title, artist, and duration in seconds (durationSeconds).
3. Duration Formatting: The Song class includes the formattedDuration() function, which converts the duration from seconds into a standardized MM:SS format using ostringstream and stream manipulators (setw, setfill).
4. Data Persistence: The MusicPlayer class utilizes private functions, loadFromFile() and saveToFile(), which read and write the playlist data to the "playlist.txt" file.
5. Formatted Display: The showPlaylist() function displays the playlist in a clean, column-aligned layout using <iomanip> functions (setw, left).
6. Dynamic Modification: Users can add new songs and delete existing songs by selection number, with automatic updates to the persistent file.

Applications of the System

This project represents core functionality relevant to library management systems. It can be applied in:

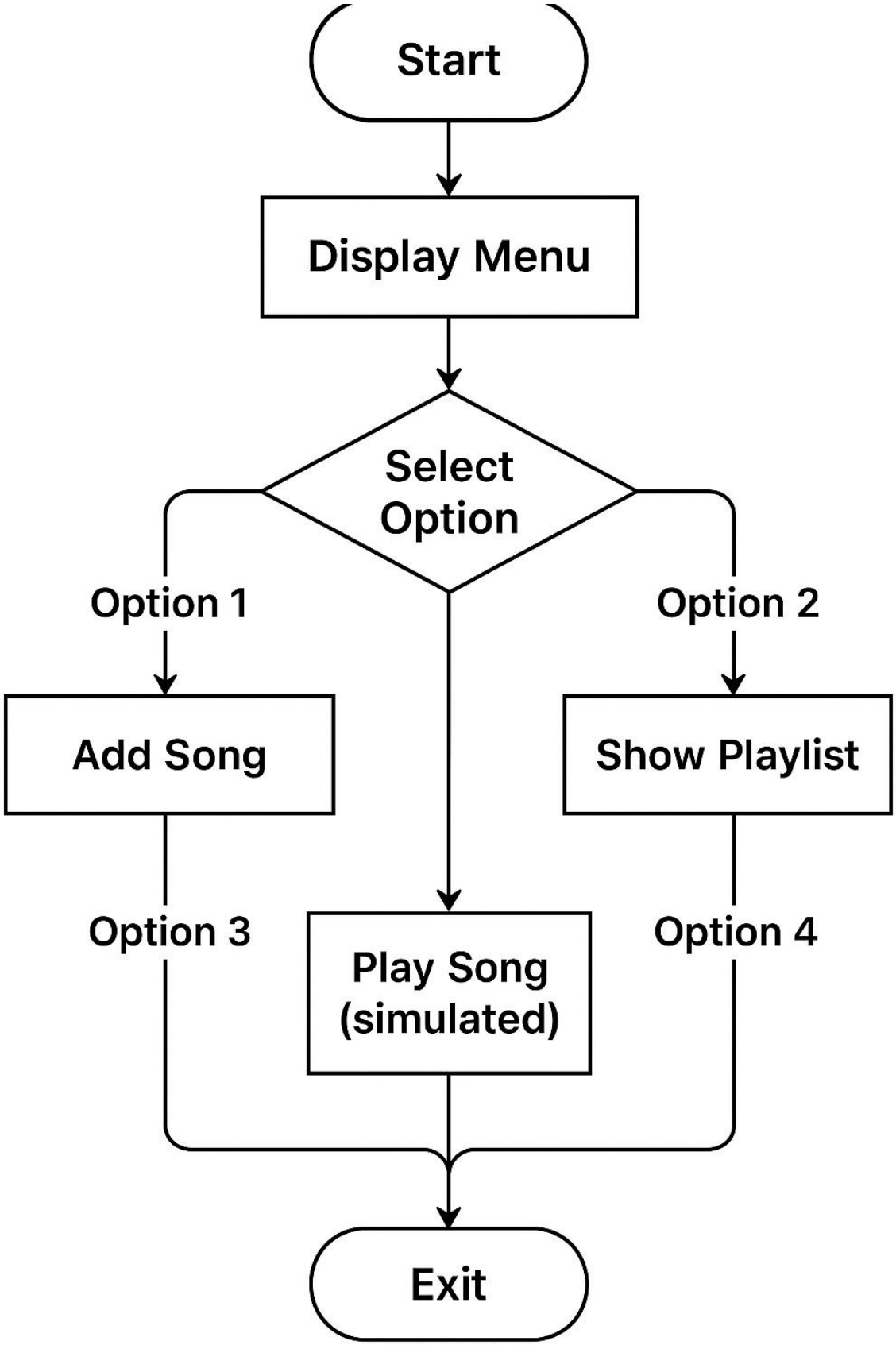
1. Learning Environments: As an educational tool to help students understand OOP, file handling, string manipulation, and I/O formatting in C++.
2. Concept Development: The project can serve as a foundation for developing more complex media player applications.
3. Basic Media Library Management: Serving as a simple, text-based system for cataloguing and managing personal media files.

Working of the System

The working of the system begins when the MusicPlayer object is instantiated in main(), which automatically calls the loadFromFile() function to retrieve existing songs from "playlist.txt".

The user is then presented with a persistent menu.

- If Add Song is selected, the user provides the title, artist, and duration (MM:SS). The duration input is parsed using stringstream into total seconds, and a new Song object is created and added to the playlist vector. The playlist is then saved immediately using saveToFile().
- If Show Playlist is selected, the system iterates through the playlist vector and displays the details of each song in a formatted table.
- If Play Song or Delete Song is selected, the system first displays the playlist and prompts for a song number. Selection logic verifies the choice before either displaying the playing song title or removing the song using playlist.erase() and updating the file via saveToFile().
- The system repeats the menu options until the user chooses option 5, which terminates the program using return 0.


Start
Display Menu
Select Option
Option 1
Option 2
Add Song
Show Playlist
Option 3
Play Song (simulated)
Option 4
Exit

System Requirements

Hardware Requirements

- A computer or laptop
- Minimum 2 GB RAM
- Basic input/output devices (keyboard and monitor)

Software Requirements

- Operating System: Windows / Linux / macOS
- Programming Language: C++
- C++ Compiler (such as GDB, or Visual Studio Code with C++ extension)

Design and Implementation

The system is designed using Object-Oriented Programming concepts in C++. The program is structured into two main classes: Song and MusicPlayer. The Song class holds the track's details (title, artist, durationSeconds). The MusicPlayer class manages the std::vector<Song> playlist and handles all file and user interactions.

OOP Concepts Applied:

- Classes and Objects: Demonstrated through the Song and MusicPlayer classes.
- Encapsulation: Data members within the Song class (like durationSeconds) are managed and accessed through member functions (like formattedDuration()). The file loading and saving logic are handled privately within the MusicPlayer class.

Implementation Details: The program utilizes the <fstream> library to handle file operations for storing playlist history in "playlist.txt," ensuring data persistence. Conditional statements and looping structures are used to manage the main menu and selection logic. The use of <iomanip> ensures a clean and readable presentation of the playlist. Input handling for duration (MM:SS) uses stringstream to parse the format correctly.

Output

main.cpp playlist.txt

```
1  Title: Midnight Calls | Artist: Harkirat Sangha | Duration: 02:43
2
```

```
===== MUSIC PLAYER MENU =====
1. Add Song
2. Show Playlist
3. Play Song
4. Delete Song
5. Exit
Enter your choice: 1

Enter Song Title: Midnight Calls
Enter Artist Name: Harkirat Sangha
Enter Duration (MM:SS): 2:43

Song Added Successfully!

===== MUSIC PLAYER MENU =====
1. Add Song
2. Show Playlist
3. Play Song
4. Delete Song
5. Exit
Enter your choice: 3

Your Playlist:
No. Title                    Artist                   Duration
------------------------------------------------------------
1   Midnight Calls           Harkirat Sangha          02:43

Enter song number to play: 1

Now Playing: Midnight Calls by Harkirat Sangha

===== MUSIC PLAYER MENU =====
1. Add Song
2. Show Playlist
3. Play Song
4. Delete Song
5. Exit
Enter your choice: 5
```

Conclusion

This project successfully demonstrates the implementation of a simple music playlist management system using the concepts of Object-Oriented Programming in C++. The program enables a user to maintain a playlist, manage individual songs, and achieve data persistence using file handling.

By utilizing classes, data encapsulation, stream manipulation, and file handling, the system reflects how basic data management applications function. The project provides practical experience in structuring and organizing code, handling complex user input (MM:SS duration), and maintaining records using external files. Through this exercise, the understanding of OOP principles becomes clearer, particularly in terms of designing classes and managing interactions between them.

Overall, the project fulfils its objectives and serves as a solid foundation for further learning and development in software design and real-world application modelling.

References

1. Documentation of C++ Standard Library: https://cplusplus.com/reference/.
2. C++ Tutorial Materials and Sample Programs from class notes.
3. Online compilation resources such as https://www.onlinegdb.com/.