

## **Project Documentation: Programming with C# - Yunus's Music Library**

**Project Description:** This project aims to simulate the management of collections of artists, albums, and songs, applying the core principles of Object-Oriented Programming (OOP). The application is a console-based program developed in C#, which uses various structures such as lists and collections to store and manage data.

The application allows the user to add artists, albums, and songs, display the stored data, view the history of operations, and provide statistics about the number of registered artists, albums, and songs. Additionally, the application allows deleting relevant data, which is done in a simple manner by removing the data and re-adding it.

### **Technical Requirements:**

1. **Programming Language:** C#
2. **Application Type:** Console-based
3. **Implementation of OOP Principles:**
  - a. **Class and Object Creation:** Classes for Artist, Album, Song, and History are created.
  - b. **Encapsulation:** Data is securely managed through encapsulation of attributes and methods.
  - c. **Inheritance:** Inheritance is used for managing the history of operations performed in the system.
  - d. **Polymorphism:** Methods are overloaded and overridden to support various operations for adding, deleting, and displaying data.
4. **Classes Created:**
  - a. **Artist:** Class for representing artists.
  - b. **Album:** Class for representing albums and linking them to artists.
  - c. **Song:** Class for representing songs and linking them to albums.
  - d. **History:** Class for recording the operations performed in the system.
5. **Implemented Methods:**
  - a. **Adding Data:** For artists, albums, and songs.
  - b. **Displaying Data:** Listing artists, albums, and songs.
  - c. **Deleting Data:** Option to delete artists, albums, or songs.
  - d. **Displaying History:** Using a list to keep track of operations performed.
  - e. **Statistics:** Displaying the number of registered artists, albums, and songs.
6. **Optional Features:**
  - a. **Saving and Reading Data:** Data is saved and loaded from JSON files.
  - b. **Statistical Reports:** Simple statistics showing the number of registered items.
  - c. **History System:** Every operation performed is logged and stored in the history list.

**Code Organization:** The code is organized into separate files for each class (e.g., Artist.cs, Album.cs, Song.cs, History.cs). The classes are implemented using OOP principles and are separated in a way that allows for easy use and future extension.

**Conclusion:** This project successfully demonstrates the use of core Object-Oriented Programming principles by implementing a console-based music management system. Through a clean class structure and encapsulated logic, the program allows users to manage a collection of artists, albums, and songs efficiently. Although there is no dedicated update function, the system handles updates by removing and re-adding entries, which maintains consistency within the collection.

The application offers interactive features such as adding new entries, listing existing ones, deleting items, and viewing the operation history. Optional features like JSON-based data saving/loading and operation logging were also implemented to enhance the functionality and reliability of the system.

Overall, the project provides a simple yet practical solution for organizing and managing music-related data.

Yunus Emre Tosun