

Assignment 1: Library Management System Using Object-Oriented Programming

CS253-2025

IIT Kanpur

Objective

Design and implement a **Library Management System (LMS)** using **Object-Oriented Programming (OOP)** concepts in **C++**. This system should demonstrate key OOP principles such as **abstraction**, **inheritance**, **polymorphism**, and **encapsulation** to manage the operations of a library effectively.

Introduction

Gone are the days when libraries were simple collections of books manually managed by librarians. We are entering an age of modern, digitized library systems where users can seamlessly interact with the system to access resources and manage borrowing operations efficiently.

In this mini-project, you will implement a **Library Management System (LMS)** in **C++** using Object-Oriented Programming concepts. This system will allow different types of users to manage their interactions with the library:

- **Students and Faculty:** Can borrow books, return them, and view their borrowing history.
- **Librarians:** Can manage books and users.

Each user and book will have unique identification numbers for tracking. Users should be able to borrow and return books within specified limits. Students will incur fines for overdue returns, while faculty members will have extended borrowing privileges without fines. Librarians will have the authority to add, remove, and update both books and users in the system.

The system will provide a clean and user-friendly interface to perform operations such as borrowing, returning, and viewing details of books and users. It will also demonstrate the implementation of OOP concepts like inheritance, abstraction, and polymorphism through its class structure.

1 Assignment Requirements

Define a User base class and derive specific roles (Student, Faculty, and Librarian) using inheritance.

1. User Roles and Permissions

- Students:
 - Can borrow up to 3 books at a time.
 - Maximum borrowing period: 15 days.
 - Fines: 10 rupees per day for overdue books.
 - Cannot manage books or other users.
- Faculty:
 - Can borrow up to 5 books at a time.
 - Maximum borrowing period: 30 days.
 - Fines: No fine for overdue books.

- Cannot manage books or other users.
- Librarian:
 - Can manage the library, including:
 - Adding, removing, or updating books.
 - Adding or removing users.
 - Can not borrow books.

2. Books

Define a Book class to represent books in the library.

Attributes:

- title, author, publisher, year, and ISBN.
- Status: Tracks whether the book is available, borrowed, or reserved.

Constraints:

- Start with at least 5 books in the system.
- Books can only be borrowed if their status is "Available."

3. Account

Create an Account class to track user activity. Each user has one account. The account must:

- Maintain a record of currently borrowed books.
- Track overdue books and calculate fines.

4. System Constraints and Rules

- (a) The system should persist its data using files. This ensures that the library's state (e.g., user records, borrowed books, and fines) is retained between program sessions.

• Borrowing Rules:

- Students and faculty can borrow books based on their role constraints.
- If a user tries to borrow more than the allowed number of books, the system should deny the request.
- If a user has unpaid fines, borrowing new books should not be allowed until the fines are cleared.
- The system should provide an option for users to simulate payment of fines.
- Users can view their total outstanding fines and mark them as paid through a dedicated menu option.
- Once the payment is made, the fine amount should reset to zero, and borrowing restrictions should be lifted.

• Returning and Updating Rules:

• Update Book Status:

Upon the return of a book, its status should be updated to "Available" in the system.

• Fine Calculation:

* For Students:

$$\text{Fine} = \text{Days Overdue} \times 10 \text{ rupees/day.}$$

* For Faculty:

- No fine for overdue books.
- Faculty members cannot borrow additional books if they (a) Have already reached the limit of 5 borrowed books, or (b) Have an overdue book for more than 60 days.

• Overdue Check:

If the book is returned after the borrowing period (15 days for students, 30 days for faculty), the system should:

- * Calculate the overdue period.
- * Display to User Side
- **User Account Update:**
 - * Remove the book from the current borrow list in the user's account.
 - * Add the book to the borrowing history.
- **Borrowing Eligibility:**
 - * If fines exist, prevent further borrowing until the fine is cleared.

2 Implementation Instructions

1. Classes:

- Create at least four classes: User, Book, Account, and Library.
- Use inheritance to derive Student, Faculty, and Librarian from the User class.

2. Encapsulation:

- Use private attributes for sensitive information like user credentials and account details.
- Provide public methods for controlled access to private attributes. Each attribute of the class must have a member function to either update. Each attribute of a class must have a member function, that prints or shows the value of the attribute. the attribute or reset the value.

3. Polymorphism: Implement common methods like borrowBook() and returnBook() differently for Student and Faculty. Use files to save data such as user details, book records, and borrowing transactions.

4. File system:

- When the program starts, load data from files to initialize the library system.
- When the program shuts down, save the current state of the library system back to files.
- Use appropriate file handling techniques to serialize and deserialize the data.

In the main code() first build the library with at least 10 books, 5 students , 3 faculty and 1 librarian. Ensure that permissions to access methods are granted based on the role of the user account in **Account** class.

The classes suggested here are just exemplary. Feel free to be a bit creative by adding your own data members or member functions such that the system is implemented.

Contact

Any queries regarding this assignment should be directed to **Gopichand Ankem** (gopichandankem@cse.iitk.ac.in) and **Nij Bharkat Kumar Padariya** (nij@cse.iitk.ac.in). Please mention CS253 in the subject line while sending any email.