

Capstone Project: Library Management System

Submission Deadline: December 17, 2024

1 Project Overview

You are tasked with creating a Library Management System using C++. This system will be used by librarians to manage book inventories, track book checkouts and returns, and handle reservations. The system should be able to notify the librarian of overdue books and calculate fines.

2 Project Objectives

Implement a console-based application that simulates a library management system. Utilize object-oriented programming principles to design your classes and interactions. Apply knowledge of data structures, file I/O, pointers, references, and exception handling in C++. The following is a suggested design that may help you implement your code. You don't have to stick to it. However, your design must cover all listed features:

3 Detailed Specifications

The program should read and write to an inventory file that contains the details of the available books in the library. The file is found on Canvas under the name '`book.inventory.csv`'.

3.1 Book Class:

- Attributes (variables):
 - ISBN
 - title
 - author
 - genre
 - publicationYear
 - isAvailable
- Methods/functions (add the necessary input parameters):
 - Constructors: Default and parameterized using initializer lists.
 - Necessary getters and setters
 - A function called `getFormattedDetails()` that returns book details in a formatted string.

3.2 Inventory Class:

- Attributes:
 - **books**: A **vector** of **pointers**
- Methods (add the necessary input parameters):
 - A function called **addBook** that adds a new book to the inventory .
 - A function **findBookByISBN** that searches for a book by ISBN and returns a pointer to it.
 - **removeBookByISBN**: Removes a book from the inventory based on its ISBN. Returns **true** if the book was found and removed, **false** otherwise.
 - **isBookAvailable**: Checks if a book with a given ISBN is available for checkout.
 - **listAvailableBooks**: Returns a list of all books that are currently available.
 - **listCheckedOutBooks**: Returns a list of all books that are currently checked out.
 - **updateBookAvailability**: Updates the availability status of a book with a given ISBN.
 - **countTotalBooks**: Returns the total number of books in the inventory.
 - **countAvailableBooks**: Returns the number of books that are currently available.

3.3 Librarian Class:

- Attributes:
 - **inventory**: A **reference** to an **Inventory** object.
- Methods:
 - Constructor that initializes the inventory reference.
 - A function called **checkoutBook** that checks out a book using its ISBN.
 - **returnBook**: Handles the return of a book by updating its status to **true**.
 - **updateInventoryAfterReturn**: Updates the inventory details after a book is returned.
 - **reserveBook**: Allows a user to reserve a book if it's currently available to checked out.
 - **cancelReservation**: Cancels an existing reservation for a book.
 - **processReservations**: Checks for available books that are reserved and notifies the users.
 - **renewBook**: Allows a user to extend the borrowing period of a book.
 - **processOverdueBooks**: Identifies overdue books and takes appropriate actions.
 - **calculateFine**: Calculates the fine for an overdue book based on the number of days late (20¢/day).
 - **addNewBook**: Adds a new book to the inventory.
 - **removeBookFromInventory**: Removes a book from the inventory.
 - **listAllBooks**: Lists all books currently in the inventory.
 - **listOverdueBooks**: Lists all books that are overdue.
 - **searchBooks**: Searches for books by title, author, or ISBN (consider inheritance).

4 UML Design

Create a detailed UML class diagram that includes all member attributes and methods, along with their visibility (public/private). Show relationships such as inheritance and compositions.

5 Main Application

Implement a user interface for the librarian to interact with the system through a menu-driven console application. The interface should provide options to manage the inventory, check out and return books, and view book details.

6 Report

Write a 2-3 page report detailing the functionality of your program. Address the following topics:

6.1 Project Overview

Provide a brief description of the system and its real-world application.

6.2 Design Decisions

Discuss key design decisions and the structure of your classes and methods.

6.3 Data Structures

Justify your choice of data structures for managing the system's data.

6.4 Object-Oriented Programming

Illustrate the use of OOP principles such as encapsulation, inheritance, and polymorphism.

6.5 Pointers and References

Explain the use of pointers and references in your system.

6.6 File I/O

Describe how your system performs file input and output operations.

6.7 Error Handling

Detail how your system handles errors and exceptions.

6.8 Testing

Explain your approach to testing the system's functionality.

6.9 Challenges and Solutions

Discuss any challenges encountered and the solutions implemented.

6.10 UML Diagram

Include and explain your UML class diagram.

6.11 Code Snippets

Provide and discuss code snippets that are crucial to your implementation.

6.12 Reflection

Reflect on your experience with the project and its impact on your understanding of C++.

7 Deliverables

- Source code files for your Library Management System.
- An executable file that can be run to demonstrate the application.
- A detailed report as described above.

8 Evaluation Criteria

- Correctness and completeness of the application.
- Use of object-oriented programming principles.
- Effective use of pointers and references.
- Code readability and adherence to C++ best practices.
- Quality and clarity of the report and the UML diagram.
- Functionality of the executable file.

This capstone project, including the accompanying report, constitutes **40%** of your final grade for the course. The project is divided into two main components for the assessment: the report and the technical implementation (source code and executable), each contributing to the project grade. The report will be evaluated for its depth of analysis, clarity, and understanding of the concepts applied, accounting for **15%** of your final grade. The technical implementation will be assessed based on functionality, code quality, and adherence to the project requirements, accounting for **25%** of your final grade. This comprehensive assessment is designed to reflect your proficiency in both theoretical knowledge and practical application of C++ programming.

This project is designed to challenge your ability to apply your C++ knowledge to a real-world problem. It is an opportunity to showcase your programming skills and your ability to think critically and solve complex problems. Answers should be submitted through Canvas on the respective assignment. Ensure that your submission includes all the required deliverables. Good luck and happy coding!