Simple OOP Library Management System

This is a simple console-based C++ Library Management System built to demonstrate basic **Object-Oriented Programming (OOP)** concepts.

It allows users to add, display, search, borrow, and return books using simple OOP tools without databases or advanced data structures.

Features

- **Book Management**: Add books with title and author. Each book tracks its availability.

- **Search Functionality**: Search for books by title.

- **Borrowing & Returning**: Borrow a book if available and return it to make it available again.

- **Availability Tracking**: Each book object maintains its own availability status.

- **User Interaction**: Console menu lets users choose operations repeatedly until exit.

Technical Highlights

- **Object-Oriented Programming (OOP)**:

- `Book` class encapsulates title, author, and availability.

- `Library` class manages a collection of Book objects and provides operations.

- Encapsulation ensures private data is only modified through public methods.

- **Data Structures**:

- Uses `std::vector` to store books during program execution.

- **Console Interaction**:

- Menu-driven loop allows continuous interaction until user exits.

Limitations

- Books are identified only by title (no unique IDs).

- Search is case-sensitive and requires exact matches.

- Data is not persistent; library is lost when the program closes.

- Does not use `std::set` or `std::map` for optimized search.

OOP Concepts Demonstrated

- **Class**: `Book`, `Library`.

- **Object**: Each book is an object instance of `Book`.

- **Attributes/Fields**: `title`, `author`, `available`.

- **Methods**: `borrowBook()`, `returnBook()`, `display()`, `addBook()`, `showAllBooks()`, `searchBook()`.

- **Constructor**: Initializes book details when created.

- **Encapsulation**: Book details are private, accessed via public methods.

Future Improvements

- Add unique IDs for books.

- Implement search by ID and partial, case-insensitive search by title.

- Add persistence using file storage.

- Extend system to track library members and borrow history.

- Optionally, create a GUI for better interaction.

This project serves as an educational starting point for learning OOP in C++.