SYNOPSIS

PROJECT TITLE:

Library Management System

(Developed using Object-Oriented Programming in C++)

---

1. Introduction

In traditional manual library systems, librarians maintain extensive paper registers to track books, issuance records, and returns. This manual process is time-consuming, prone to human error (such as misplaced records or illegible handwriting), and inefficient when searching for the status of a specific book.

The "Library Management System" mini-project is a software application designed to automate the basic operations of a library. It utilizes the power of computer data storage to replace paper records with digital files. The system is developed using the C++ programming language, employing Object-Oriented Programming (OOP) concepts to model real-world library entities like books.

2. Problem Definition

The primary problem this project addresses is the inefficiency of manual record-keeping in small to medium-sized libraries. Specific issues include:

- Difficulty in quickly determining if a book is available on the shelf or currently issued to someone.
- Risk of data loss if physical registers are damaged.
- Time wasted manually writing down book details repeatedly during issuance and return.

3. Objectives of the Project

The main objective is to develop a console-based application that streamlines library operations. Specific objectives include:

- **Data Persistence:** To ensure that book records are saved permanently in a file on the disk, so data is not lost when the program is closed.
- **Efficient Record Keeping:** To allow the librarian to easily add new book records containing details like ID, Title, and Author.
- **Status Tracking:** To maintain a real-time status flag for every book (Issued vs. Available).
- **Automated Transactions:** To provide simple mechanisms to "Issue" and "Return" books, which automatically update the book's status in the file.

4. Project Scope and Limitations

Scope: This project covers the essential backend operations required to manage book inventory and circulation in a small library setup. It focuses on the interaction between the librarian and the book database.

Limitations: As this is a mini-project developed under specific constraints (Turbo C++ compiler), it has certain limitations:

- **Single User:** The system does not have separate logins for librarians and students. It is intended for use by library staff only.
- **No Fine Calculation:** It does not track due dates or calculate late return fines.
- **Console Interface:** It uses a text-based DOS interface rather than a modern Graphical User Interface (GUI).

5. Methodology and Technical Approach

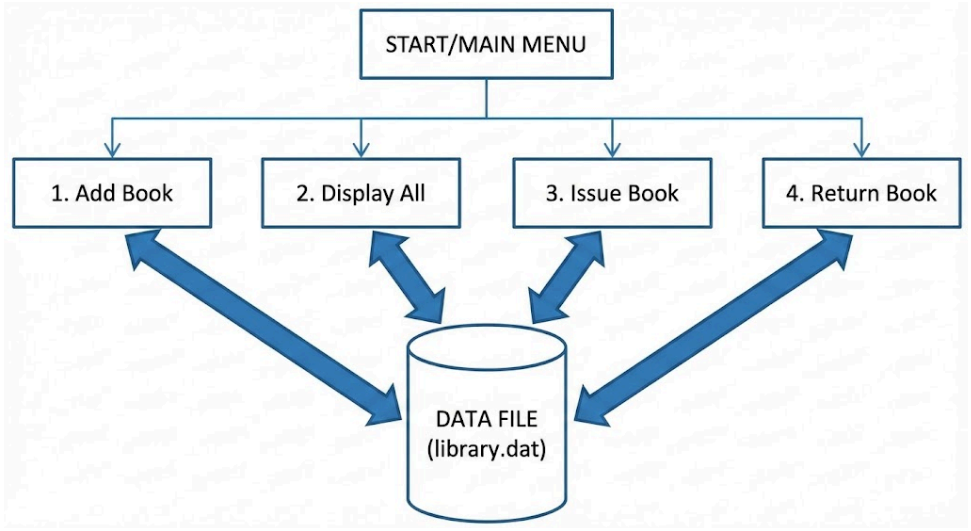
The project is built using the C++ programming language on the Turbo C++ IDE platform. The core methodology relies on two key concepts:

A. Object-Oriented Programming (OOP): We utilize a class named Book. This class encapsulates the data (properties like bookID, title, author, isIssued) and the functions that operate on that data (createBook, showBook). This keeps the code organized and modular.

B. File Handling: To achieve data persistence, the program uses C++ binary file handling (`fstream`).

- Data is stored in a binary file named library.dat.
- The program reads and writes complete objects of the Book class to this file.
- Functions like seekp() are used during issuing and returning to locate a specific book record in the file and update just that record without rewriting the whole file.

6. System Flow Diagram



7. Hardware and Software Requirements

Software Requirements:

- **Operating System:** Windows
- **Development Environment (IDE):** Turbo C++ 3.0 or compatible older compilers.
- **Language:** C++

Hardware Requirements (Minimal):

- Intel Pentium Processor or higher.
- 128 MB RAM.
- 10 MB free hard disk space for the compiler and data files.

8. Implementation Details (Snapshots)

[Instruction: After you run the code in Turbo C++, take screenshots of the output screens and paste them here.]

Figure 8.1: Main Menu Screen

[Insert Screenshot of the Main Menu showing options 1-5]

Figure 8.2: Adding a New Book Record

[Insert Screenshot showing the user entering Book ID, Title, and Author]

Figure 8.3: Displaying All Books (showing status)

[Insert Screenshot showing the table listing books and their Issued/Available status]

9. Conclusion

The proposed Library Management System successfully demonstrates the application of C++ and file handling concepts to solve a real-world problem. It provides a functional, albeit basic, tool for managing library records digitally, ensuring data safety and quicker transaction processing compared to manual methods. It serves as an excellent foundation that can be expanded in the future with features like student management and fine calculation.