Library Management System

Project Goal:

Create a basic library management system that allows users to add, remove, and search for books.

Required Concepts:

- **Functions**: Define functions for adding, removing, and searching for books.
- **Classes and Objects:** Create a Book class to represent books with attributes like title, author, ISBN, and availability.
- **Static Members**: Use static members to keep track of the total number of books in the library.
- **Constants:** Define constants for maximum book capacity or other library-specific values.
- **Operator Overloading:** Overload operators like == and < for comparing books based on title, author, or ISBN.
- **Type Conversion:** Implement type conversion to allow for easier comparison and sorting of book objects.
- **Pointers:** Use pointers to dynamically allocate memory for the book objects.
- **Arithmetic:** Perform arithmetic operations on book-related data (e.g., calculating the number of available books).
- **Dynamic Memory**: Use new and delete to allocate and deallocate memory for book
- **Functionality:** The system should accurately add, remove, and search for books.
- **Data Structures:** Use classes and objects effectively to represent books and library data.
- **Memory Management:** Correctly allocate and deallocate memory using pointers and dynamic memory.
- **Code Organization:** The code should be well-structured, modular, and easy to understand.

- **Error Handling:** Implement appropriate error handling to prevent unexpected behavior.
- **Efficiency:** The system should perform efficiently, especially for large libraries.
- **User Interface:** Provide a basic user interface for interacting with the system.
- **Inheritance**: Create a derived class for digital books if applicable, inheriting properties from the base Book class.
- **Polymorphism:** Use polymorphism to handle different types of books (e.g., physical and digital) in a uniform way.
- **Strings:** Use strings to represent book titles, authors, and ISBNs.
- **Exception Handling:** Implement exception handling to gracefully handle errors like invalid input or out-of-memory conditions.