Design a C++ class named Book with the following attributes:
title (string)
author (string)
isbn (string)
available (boolean)
Design a class named Library with the following attributes:
books (an array or vector of Book objects)
Methods:
addBook(book): Adds a new book to the library.
searchBookByTitle(title): Searches for a book based on its title.
borrowBook(isbn): Marks a book as unavailable if it's available.
returnBook(isbn): Marks a book as available.

C++ Implementation

```
#include <iostream>
#include <vector>
#include <string>

class Book {
public:
    std::string title;
    std::string author;
    std::string isbn;
    bool available;

    // Constructor
    Book(const std::string& t, const std::string& a, const std::string& i)
        : title(t), author(a), isbn(i), available(true) {}

    // Display book details
    void display() const {
        std::cout << "Title: " << title << "\nAuthor: " << author
                  << "\nISBN: " << isbn << "\nStatus: "
                  << (available ? "Available" : "Checked Out") << "\n";
    }
};

class Library {
private:
    std::vector<Book> books;
```

```
public:
    // Add a book to the library
    void addBook(const Book& book) {
        books.push_back(book);
        std::cout << "Book added: " << book.title << "\n";
    }

    // Search for a book by title
    Book* searchBookByTitle(const std::string& title) {
        for (auto& book : books) {
            if (book.title == title) {
                return &book;
            }
        }
        return nullptr;
    }

    // Borrow a book using ISBN
    void borrowBook(const std::string& isbn) {
        for (auto& book : books) {
            if (book.isbn == isbn) {
                if (book.available) {
                    book.available = false;
                    std::cout << "You have borrowed: " << book.title << "\n";
                } else {
                    std::cout << "Sorry, this book is already checked out.\n";
                }
                return;
            }
        }
        std::cout << "Book with ISBN " << isbn << " not found.\n";
    }

    // Return a book using ISBN
    void returnBook(const std::string& isbn) {
        for (auto& book : books) {
            if (book.isbn == isbn) {
                if (!book.available) {
                    book.available = true;
                    std::cout << "You have returned: " << book.title << "\n";
                } else {
                    std::cout << "This book is already available in the library.\n";
                }
                return;
```

```
            }
        }
        std::cout << "Book with ISBN " << isbn << " not found.\n";
    }

    // Display all books in the library
    void displayAllBooks() const {
        if (books.empty()) {
            std::cout << "No books in the library.\n";
            return;
        }
        for (const auto& book : books) {
            book.display();
            std::cout << "----------\n";
        }
    }
};

int main() {
    Library myLibrary;
    int choice;
    std::string title, author, isbn;

    do {
        std::cout << "\nLibrary Menu:\n";
        std::cout << "1. Add Book\n2. Search Book by Title\n3. Borrow Book\n4. Return Book\n5.
Display All Books\n6. Exit\n";
        std::cout << "Enter choice: ";
        std::cin >> choice;
        std::cin.ignore();  // Clear newline from buffer

        switch (choice) {
            case 1:
                std::cout << "Enter title: ";
                std::getline(std::cin, title);
                std::cout << "Enter author: ";
                std::getline(std::cin, author);
                std::cout << "Enter ISBN: ";
                std::getline(std::cin, isbn);
                myLibrary.addBook(Book(title, author, isbn));
                break;

            case 2:
                std::cout << "Enter title to search: ";
```

```
            std::getline(std::cin, title);
            if (Book* book = myLibrary.searchBookByTitle(title)) {
                book->display();
            } else {
                std::cout << "Book not found.\n";
            }
            break;

        case 3:
            std::cout << "Enter ISBN to borrow: ";
            std::getline(std::cin, isbn);
            myLibrary.borrowBook(isbn);
            break;

        case 4:
            std::cout << "Enter ISBN to return: ";
            std::getline(std::cin, isbn);
            myLibrary.returnBook(isbn);
            break;

        case 5:
            myLibrary.displayAllBooks();
            break;

        case 6:
            std::cout << "Exiting program...\n";
            break;

        default:
            std::cout << "Invalid choice. Try again.\n";
      }
   } while (choice != 6);

   return 0;
}
```