
PROJECT COMMANDLINE BOOKSTORE

PREPARED BY

Sana Fatima

PREPARED TO

A&D Tech

sana.faati@gmail.com

ad.techinnov25@gmail.com



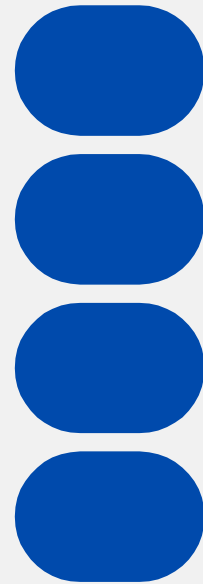
TABLE OF CONTENTS

Problem Statement

Solution Overview

Challenges and Resolution

Instructions



Problem Statement

Implementing the Program

- Create a functional bookstore system in the command line that allows users to browse books, add them to a shopping cart, and manage purchases.
- Design Key Components:
- Book Class:
 - Attributes: ISBN, title, author, price, number of copies available.
 - Methods: update inventory, display book info.

- **Cart Class:**
 - Attributes: list of books (with quantities), total cost.
 - Methods: add book, remove book, calculate total.
- **Store Class:**
 - Attributes: inventory (a list or array of books).
 - Methods: search book by ISBN, title, or author, display available books, check out (finalize purchase and update inventory).

SOLUTION OVERVIEW

The solution overview for this program is:

The system includes functionality for managing a store's inventory of books, a shopping cart, and processing customer transactions. The code is divided into three main classes: Book, Cart, and Store. The Book class represents individual books with attributes such as ISBN, title, author, price, and the number of copies. The Cart class manages the shopping cart operations, including adding, removing, and calculating the total cost of books. The Store class manages the store's inventory, including adding books, searching for books by ISBN, displaying available books, and processing checkout operations.

CHALLENGES AND RESOLUTIONS

1. Error Handling for User Input:

- **Challenge:** Handling invalid user input, such as incorrect menu choices or invalid data types, is crucial to ensure the program doesn't crash or behave unexpectedly.
- **Resolution:** The program includes error checking using `cin.clear()` and `cin.ignore()` to handle invalid input types and ensure users enter valid choices for menu options and data fields.

2. Cart Management:

- **Challenge:** Managing books in the cart, including adding, removing, and updating the total cost.

- **Resolution:** The Cart class provides methods to add and remove books, with the total cost being updated accordingly. However, the Cart class assumes a fixedsize array (Book b[100]), which may limit scalability. A dynamic array could be used for better flexibility.

3. Checkout Process:

- **Challenge:** Ensuring that the checkout process accurately updates the inventory and handles edge cases, such as empty carts or outofstock books.
- **Resolution:** The Store class's checkOut method handles reducing the inventory count for each book in the cart. It checks for stock availability and prints errors if the cart is empty or if any book is out of stock.

INSTRUCTIONS

1. Compile and Run the Program:

- Ensure that all classes (Book.h, Cart.h, Store.h) include appropriate header files.
- Compile the program using a C++ compiler (i.e dev++, online gdb)

2. Program Interaction:

- Upon running the program, a menu will be displayed with options to view books, add books to the cart, remove books from the cart, checkout, add a new book to the inventory, or exit.
- Follow the prompts to perform each action. Input validation will ensure that only valid choices and data types are accepted.

3. Adding Books to Inventory:

- To add a new book, select option 5 and follow the prompts to enter the book's ISBN, title, author, price, and number of copies.
- The program checks for duplicate ISBNs to prevent adding the same book twice.

4. Cart Operations:

- To add a book to the cart, select option 2 and enter the book's ISBN. If the book is available, it will be added to the cart.
- To remove a book from the cart, select option 3 and enter the book's ISBN.

5. Checkout:

- To proceed with checkout, select option 4. The program will update the store's inventory based on the books in the cart.
- If the cart is empty or if any book is out of stock, appropriate error messages will be displayed.

6. Exiting the Program:

- To exit, select option 6. The program will terminate gracefully.

