

ECE 36800 – Data Structures Programming Assignment 2

Purpose of Assignment 2:

Design and implement a bookstore user interactive program.

Goal of Part 2:

Create a simple bookstore customer interface program, allow customers to browse through the catalog, add books to the cart, remove books from the cart, and checkout books.

What to submit:

1. `booklist2.cpp`: The implementation file for the new booklist class. Start with the given file and add your implementation. Include implementations from your `booklist1.cpp` from Part 1 and add implementations of the new member functions. If your `booklist1.cpp` has errors, please correct those errors first.
2. `storeUI.cpp`: The application program handling user interaction. Start with the given file and add your implementation. Include implementations from your `catalog.cpp` from Part 1 and add new contents to handle the customer requests. If your `catalog.cpp` has errors, please correct those errors first.
3. A word document `proj2p2.docx`: This file should include the printout of your program with different user interaction cases. Make sure you have sufficient test cases to prove the correctness of your program. Include a copy (or screenshot) of the printout of the program run for each test case.
4. Push all your files under the “proj2/part2” directory before the deadline.

Other available files (please do not modify):

1. `book.h` and `book.cpp`: The implementation file for the book class.
2. `booklist2.h`: The header file for the new booklist class, which is implemented using a singly linked list.
3. `booklist.txt`: A list of books along with their features, each delimited by a TAB.
4. `Makefile`: Rules to compile the source files for your convenience.

Note: The following files `book.h`, `book.cpp`, and `booklist.txt` are exactly the same as those included in Part 1.

Guideline:

After the program starts, it first read from `booklist.txt`, and stores the books in a linked list called `catalog`. Next, the program prints the following menu and then waits for user input. Customers can print the book catalog and add or remove books to/from the customer's `cart`. Finally, a customer can check out the selected books in the cart.

Menu:

```
[L]ist all books  
[A]dd a book into cart  
[R]emove a book from cart  
[C]heckout  
[Q]uit
```

- If the user input is “L” or “l”, the content of the current book catalog will be printed.
- If the user input is “A” or “a”, the user will be asked to enter the book id and the number of copies to be added into the cart.
- If the user input is “R” or “r”, the user will be asked to enter the book id and the number of copies to be removed from the cart.
- If the user input is “C” or “c”, the system will print the content in the cart and calculate the total price and exit the program successfully.
- If the user input is “Q” or “q”, the program will exit successfully.

A sample execution of the program is shown in the end of this document to illustrate the basic functionality of the bookstore database (bolded texts are user inputs).

Explanations of Implementation:

1. User's `cart` is defined as a `booklist` object.
2. Please note that when a certain number of copies of a book are added to your cart, the same number of copies of that book should be removed from the catalog – so the total number of copies remains the same. In the same manner, when a certain number of copies of a book are removed from your cart, the same number of copies of that book should be added back to the catalog. In addition, when a customer adds all remaining copies of a book from the catalog into the cart, that book record in the catalog should be removed from the catalog. Similarly, when all copies of a book are removed from the cart and put back to the catalog, this book record should be removed from the cart.
3. The same book can be added/removed multiple times to/from the cart. However, the cart content should only show one entry of the same book. For example, if a customer adds one copy of book 1 the first time, and later on decides to add 2 more copies of book 1 again, the cart content should show 3 copies of book 1, instead of two separate entries. Similarly, when the catalog is printed, each book should only have one consolidated entry (showing the current number of copies).

Grading Policy:

Part 2 counts for 60% of the overall points in Programming Assignment 2.

Please make sure your program compiles successfully. If not, 10 points will be deducted.

1. **Executability** (5%):

- Runtime errors: You program must not have runtime errors (e.g., code crash, infinite loop, reading uninitialized memory, accessing the content of a NULL pointer, etc.).

2. **Programming style** (5%):

- Code efficiency: Code should use the best approach in every case.
- Readability: Code should be clean, understandable, and well-organized. Please pay special attention to indentation, use of whitespace, variable naming, and organization.
- Documentation: Code should be well-commented with file header and comments.

3. **Program Specifications/Correctness** (50%):

Please refer to the Grading Criteria table for details. Specifically, your program should behave correctly, adhere to the instructions, and pass the test program.

file	item	weight (%)
booklist2.cpp	add_book_copies	7
	remove_book_copies	7
	calculate_total_price	7
storeUI.cpp	Add	7
	Remove	7
Correct implementation from part 1		5
proj2p2.docx	running result	10
total		50

Example program execution:

Welcome to BookSea

```
=====
Menu:
[L]ist all books
[A]dd a book to cart
[R]emove a book from cart
[C]heckout
[Q]uit
=====
```

Command: **L**

Book Title	Price	Copies
-----	-----	-----
1. Lisey's Story	\$28.00	10
2. The March	\$14.95	10
3. The Lincoln Lawyer	\$26.95	10
4. The Morning Star	\$17.00	10
5. Tuesdays with Morrie	\$6.99	15
6. The Da Vinci Code	\$7.99	15
7. The Rings Book	\$24.95	12
8. Following in Lincoln's Footsteps	\$16.00	8
9. The Lord of the Rings	\$35.00	10
10. The Perfect Storm	\$14.00	12

```
=====
Menu:
[L]ist all books
[A]dd a book to cart
[R]emove a book from cart
[C]heckout
[Q]uit
=====
```

Command: **A**

Book ID? **1**
How many copies? **1**

Items in cart:

Book Title	Price	Copies
-----	-----	-----
1. Lisey's Story	\$28.00	1

```
=====
Menu:
[L]ist all books
[A]dd a book to cart
[R]emove a book from cart
[C]heckout
[Q]uit
=====
```

Command: **A**

Book ID? **3**
Copies? **2**

Items in cart:

Book Title	Price	Copies
-----	-----	-----
1. Lisey's Story	\$28.00	1
3. The Lincoln Lawyer	\$21.56	2

Command: **A**
 Book ID? **5**
 How many copies? **1**

Items in cart:

Book Title	Price	Copies
-----	-----	-----
1. Lisey's Story	\$28.00	1
3. The Lincoln Lawyer	\$21.56	2
5. Tuesdays with Morrie	\$6.99	1

=====
 Menu:
 [L]ist all books
 [A]dd a book to cart
 [R]emove a book from cart
 [C]heckout
 [Q]uit
 =====

Command: **L**

Book Title	Price	Copies
-----	-----	-----
1. Lisey's Story	\$28.00	9
2. The March	\$14.95	10
3. The Lincoln Lawyer	\$26.95	8
4. The Morning Star	\$17.00	10
5. Tuesdays with Morrie	\$6.99	14
6. The Da Vinci Code	\$7.99	15
7. The Rings Book	\$24.95	12
8. Following in Lincoln's Footsteps	\$16.00	8
9. The Lord of the Rings	\$35.00	10
10. The Perfect Storm	\$14.00	12

=====
 Menu:
 [L]ist all books
 [A]dd a book to cart
 [R]emove a book from cart
 [C]heckout
 [Q]uit
 =====

Command: **R**
 Book ID? **1**
 How many copies: **1**

Items in cart:

Book Title	Price	Copies
-----	-----	-----
3. The Lincoln Lawyer	\$26.95	2
5. Tuesdays with Morrie	\$6.99	1

```
=====
Menu:
[L]ist all books
[A]dd a book to cart
[R]emove a book from cart
[C]heckout
[Q]uit
=====
```

Command: **C**

Items in cart:

Book Title	Price	Copies
-----	-----	-----
3. The Lincoln Lawyer	\$26.95	2
5. Tuesdays with Morrie	\$6.99	1

Total price: \$60.89

Thanks for shopping with BookSea!

Goodbye!