

Programming Assignment #2

Bookstore Inventory

CS 2308.003, Fall 2011

Instructor: Jill Seaman

Due: in class **Thursday, 9/22/2011** (upload by **10:00am Thursday 9/22/2011**)

Problem:

Write a C++ program that will allow a user to manage the inventory of a bookstore.

Input: The inventory for a bookstore will be stored in a file. For each book (title) in the inventory, there is one line in the input file. The line contains the following information, separated by a space.

SKU (an integer)
isbn (10 digits, last one may be "X")
quantity (number of copies in inventory)
price (dollars and cents, no dollar sign)
title (may contain spaces in it, but no more than 80 characters)
The input file will be named "inventory_data.txt".

Note: You may assume the bookstore has less than 10,000 titles.

Processing: Once the inventory has been input by the program, it should offer the user a menu with the following options:

1. Add a book to the inventory.
2. Remove a book from the inventory (by sku).
3. Display the information for a book (given the sku).
4. Display the inventory sorted by sku.
5. Quit (automatically saves the inventory).

The program should perform the operation selected by number and then re-display the menu.

Do not change the menu numbers associated with the operations.

Output: When the user quits the menu, the inventory is saved to a file named inventory_output.txt. It should be saved in the same format as the input. One should be able to rename the output file to inventory_data.txt, and run the program again (so it inputs from that file) without any errors.

NOTES:

- This program does not need to be done in a Unix environment. You may use whatever C++ programming environment (Visual C++, Dev-C++, etc.) you prefer.
- The program must be modular, with significant work done by functions. Each function should perform a single, well-defined task.
- Use an array of structures to store the inventory in the program.
- Please use the code from the book in chapter 5 (Program 5-8, p 268) as a template for the menu portion of the program. If you have the disk from the book, you can cut and paste and modify the code from the disk. Note: I recommend putting the code for each "case" within curly braces to avoid certain potential compiler errors.
- For the search and the sort, you may use any of the search and sort algorithms that we studied in class. You can cut and paste and modify the code from either the disk from the book, or from the lecture pdfs on the class website.

Logistics:

Include the following header at the top of the source code file (add your filename, name, and date).

```
// File Name:  
//  
// Author:  
// Date:  
// Assignment Number: 2  
// CS 2308.004 Fall 2011  
// Instructor: Jill Seaman  
//  
// <Brief description of the contents and purpose>
```

Name your file **assign2_XXXXXXXXX.cpp** where XXXXXXXXXXX is your 9 character TX state ID number, the one that is on your ID card. It should look something like this: A04123456. If yours is just six digits, then add "A00" to the front.

There are **two** steps to the turn-in process:

1. Submit an **electronic copy** using the following upload link:
<http://www.cs.txstate.edu/~js236/homework>
(the link is also on the class webpage).
Click on CS2308.003, and log in with your Net ID and follow the directions to

upload your file.

2. Submit a **printout** of the file at the beginning of class when the assignment is due. Please print your name on the front page (stapled together, please, if you have more than one page of output).