

Programming Assignment #5

Bookstore Inventory Redux

CS 2308.003, Fall 2011

Instructor: Jill Seaman

Due: in class Thursday, 11/03/2011 (upload electronic copy by 10am)

Problem:

Write C++ Classes that will manage the inventory of a bookstore.

Book: For each book, you should store the following info:

sku	(int)	Greater than 0
quantity	(int)	0 or more
cost	(float)	0 or more
title	(a string)	Should not be empty

Note: You should NOT assume anything special about the sku. Books with different titles may have the same sku. Books with different skus may have the same title.

Bookstore Inventory:

You should be able to store 100 books. An attempt to add a book to the inventory when it already has 100 books should fail.

You should implement the following operations over the bookstore inventory:

addBook: takes a book and adds it to the inventory (unless it's full). Returns true if it succeeded.

removeBook: takes a book and removes ALL matching entries for that book from the inventory. Returns the number of books that were removed.

showInventory: displays a listing of the book inventory to the screen, nicely formatted, one book entry per line. Output sku, then quantity, then cost, then title.

sortInventory: reorders the books in the list, using the < (or >) operator over the books (does not display them).

getTotalInventory: returns the total quantity of (copies of) books in the inventory.

getCostOfTotalInventory: returns the total cost of ALL of the books in the inventory.

Classes:

- Create classes for Book and BookInventory with appropriate header files.
- Implement methods in the BookInventory class to complete the operations described above.
- You may add other private, helper, functions if you want.
- Implement functions in the Book class to:
 - set and get all instance variables
 - overload == (used to remove a book), <, and > operators (used in sorting)
 - for < and > use sku, and when the skus are the same, use the title.
 - for ==, two Books are equal if all four instance variables are equal.
- You should implement two constructors for the Book class: one that takes no arguments (sku is -1, quantity and cost are 0, title is empty), and one that takes a value for each of the member variables..

Input/Output:

The main function should be a driver program that tests the functionality of the Book and BookInventory classes. See the website for a sample driver program that should work with your code. You should write and submit your own driver which should do more complete testing.

Do not add extra I/O to the class functions. All the testing should happen in the driver.

NOTES:

This program DOES need to be done in a Linux/Unix environment. Create and use a makefile to compile the executable program. There will be four goals in this makefile, because you will have three .cpp files. Use the following names for your files:

Book.h
Book.cpp
BookInventory.h
BookInventory.cpp
BookDriver.cpp

Put a header comment at the top of each file.

DO NOT change the names of the functions or files.

Follow the rest of the style guidelines described here:
<http://www.cs.txstate.edu/~js236/styleguidelines.txt>
(there is a link to these on the course webpage as well).

Logistics:

Since there are multiple files for this assignment, we want you to combine them into one file before submitting them. You should use the zip utility from the Linux/Unix command line:

```
[...]$zip assign5_Axxxxxxxx.zip BookDriver.cpp BookInventory.cpp  
BookInventory.h Book.cpp Book.h makefile
```

This combines the 6 files into one zip file, assign5_Axxxxxxxx.zip. Then you should submit only assign5_Axxxxxxxx.zip.

Name your file **assign5_xxxxxxxxx.zip** where xxxxxxxx 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 (also see the Assignment Turn-in Policy):

1. Submit an **electronic copy** using the following upload link:

<http://www.cs.txstate.edu/~js236/homework>

(There is a link directly to this page on the course website).

Click on your course number, 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 on the day it is due. Please print your name on the front page, and staple if there is more than one page.