# University of Washington Bothell

## CSS 502: Data Structures and Object-Oriented Programming II

Winter 2016

Program 4: Nile Book Store

**Purpose**

This program will re-enforce good factoring and program design. Class inheritance, abstract classes and polymorphism is required. The usage of hashing will also be required.

**The Program**

The program will simulate the activities of a used book store. Customers and inventory will be tracked as they buy and trade assorted types of books. Three files will be read into the program are arg1, arg2 and arg3.

1) Inventory.txt
2) customers.txt
3) commands.txt

**The Books**

The book store will carry both new and used books. Different types of books require different types of data to be stored. Assume that all titles are unique within in a book type; but books can be used or new.

Details are as follows:

| | |
|---|---|
| Book: | Title, Author, Cost |
| Audio Book: | Title, Author, Cost, Narrator |
| Graphic Novel: | Title, Author, Cost, Artist |

In addition, each of the above types can be either used or new. All new books are similar but the used items require a quality designation of either: Fair, Good, or Excellent.

As titles are unique, so books are sorted by Title. An audio book may have the same title as a regular book or a graphic novel and should be treated differently.

The first file presented to the program will be the inventory file. Its format will be as follows with each input separated by a comma. Assume no errors in the input file.

Single Letter: denoting what type of book (B: Book, A: Audio Book, G: Graphic Novel)
Single Letter: denoting Used or New (U: Used, N: New)
Integer: Denoting number of items

string(s): Author’s name
string(s): Title
float: Cost of the book
If G, string(s): Denoting the Artist’s Name
If A, string(s): Denoting the Narrator’s Name
If U, single letter denoting quality (F, G, E)

Examples:

B,N,6,William Vollmann,The Dying Grass,35.99
G,U,1,Harvey Pekar,American Splendor 4,10.23,Crumb,G
A,U,2,J.K. Rowling,Harry Potter and the Chamber of Secrets,45.45,Jim Dale,G

**The Customers**

There are multiple levels of customer which get different privileges and also may have different data stored with each:

Customer: First name, Last name
Gold Customer: First name, Last name, 10% discount on all purchases except audio books
Senior Customer: First name, last name, age, 15% discount on all purchases

Each customer will have a unique first and last name combination. Note that you will be required to keep the purchase history of each customer. A customer moves to gold status when they have spent more than 200 dollars. A customer over 65 years old gets senior status. The input file for customers will look like this:

FirstName,LastName,Age

You can assume that all customer start with a zero balance. Also assume no errors in the input of this file.

**The Transactions**

The third file passed in as an argument will be the set of transactions to conduct on the bookstore. The following set of transactions are supported:

Purchase (P): Buy a book

Return (R): Return a book that was bought

Trade-In (T): Bring in a new book(s) to add to the Inventory

History (H): Print out customer history, including any discounts and money spent

Inventory (I): Print out the inventory of the store including money in the till)

The transactions will start with the appropriate letter followed by the customer, followed by the appropriate book information. The customer has a first and last name specified separately. In particular the book will be described in the same way as the above w/o cost (except for trade-ins) and number. Assume one item is bought per line. For a trade-in assume that the book is used and also that the cost will be given. The cost for a used item will be the cost that is paid out as well as what will be charged. So if it is the same as a book in stock it should be priced appropriately.

```
P,Jim,Jones,B,N,William Vollmann,The Dying Grass
T,Holly,McHenry,A,J.K. Rowling,Harry Potter and the Chamber of Secrets,49.34,Jim Dale,E
R,Jim,Jones,B,N,William Vollmann,The Dying Grass
H,Jim,Jones
```

Assume no errors in the syntax of the commands. However, on must make sure that returns are valid before they are accepted and books are in stock before they are sold. Also, a return may knock a customer from Gold back to a regular customer.

For H, print out the purchasing history of the customer in chronological order as well as their customer status and current dollars spent. For I, print out the inventory of the store in the sort order described above.

**What to Turn In (in a .zip file):**

- All .h and .cpp files required to build your Book Store App
- A Build script which works on Linux to build the Book Store App
- A design document clearly showing the design for your store

**Other Notes**

1) Design will be a larger part of the grade for this program than past programs. Your design document should include at least the following:
    a. Short description/overview of program
    b. Class Diagram

c. Memory diagram: This is a diagram showing the use of memory (with respect to data structures) in the program. Show, for example, any linked lists, trees, graphs, and/or hash tables and any relationships between the memory used for them.

2) The program should utilize inheritance to model books and customers. The program should also take advantage of polymorphism where appropriate.
3) There should be a BookStore Class in your design and implementation. This should be full realized.
4) You will be required to use at least one Hash in this program that you implement.