

John Krueger
Project 1 – Bookstore
Features Description

For my bookstore, I implemented a total of 5 functions (however there is some overlap between some of them). I implemented the ability to add items to a cart and place an order, the ability to add new members (both premium and regular), the ability to view members, the ability to add products to inventory (books, DVDs and CDs), and the ability to view members.

For the purchasing function, I take the user through prompts that show what products are in the store's inventory, and allow them to add them to an order. The inventory of the store is stored as three ArrayList fields in the Store object; one list each for books, DVDs and CDs. These lists are accessed by the getBooks(), getDVDs(), and getCDs() methods, which return the respective list. The items in the in-progress order are stored in three ArrayLists in the TestHarness class by copying the objects from the inventory ArrayLists.

After the desired items of each type are selected and added to these lists for the in-progress order, the user is asked for his/her name. A check is then run using the findPremiumMember() and findMember() methods to see if the user is a member already. These methods return -1 if a name wasn't found in the respective list, otherwise the index of the user in the list which is used to get his/her card number (if a premium member) or update his/her total spent field. If the user is a premium member, s/he is greeted by name, his/her on file card number is used, and his/her total money spent field is updated. If s/he is a member, s/he is greeted by name, asked for a card number, and his/her total money spent field is updated. If s/he is not a member, a card number is asked for.

After determining whether a not the user is a member and getting a card number as appropriate, the card number as well as the three ArrayLists containing the contents of the user's order are passed to the store's placeOrder() method, which removes the items in the order from the store's inventory and prints that a "payment" was made using the passed card number.

The functions that add new members and add products to inventory are very similar. Each takes the user through a series of prompts to choose what type of entry should be added (a premium vs regular member or a book vs DVD vs CD) and then asks for the information for each field appropriately.

When adding a new member, the user first selects a regular or premium member; if a regular member is selected, the user is asked for a name which is used to create a new Member object which is passed to the store's addMember() method. If a premium user is selected, the user is asked for a name and card number which are used to create the PremiumMember object that is passed to the store's addPremiumMember() method.

If a book/DVD/CD is selected, the user is prompted for the name, price and author/studio/artist which is then used to create the item (using its object's constructor) which is passed to the addBook(), addDVD() or addCD() method.

Finally, the application has two more functions that display the list of registered members (regular and premium) with their information, as well as all of the books, DVDs, and CDs currently in stock. This is done using the getMembers(), getPremiumMembers(), getBooks(), getDVDs() and getCDs() methods, which return ArrayLists for each type of object. Using a for loop, information about each object is then output for the user to see.