Assignment 3
Bookstore Application 1
CS190: Java
out: 28 January 2008
due: 4 February 2008

1 Description

We will be designing, implementing and testing a bookstore application in the next several homework assignments. Each assignment will build on the previous. It will be to your advantage to complete each assignment and do so on-time.

2 Implementation Instructions

Create a java object called Merchandise which will contain the following information:
- price (double)
- SKU (int)
- title (String)

The object should have the following behaviors (methods) implemented:
- double getPrice()
- void setPrice(double price)
- int getSKU()
- void setSKU(int SKU)
- String getTitle()
- void setTitle(String title)

Create a java object called Book which will contain the following information:
- editor (String)
- genre (String)

The object should have the following behaviors (methods) implemented:
- String getEditor()
- void setEditor(String name)
- String getGenre()
- void setGenre(String genre)

The object Book will extend the object Merchandise. Use the extends keywords to signify this relationship in the class declaration for the Book object.

    public class Book extends Merchandise
Create a java object called Inventory which will contain the following information:

- merchandise (Merchandise)
- available (int)

The merchandise object contained within the Inventory means that that item exists in the product catalog. The available property refers to the actual number in stock. It is possible to have merchandise set, but have the available as 0.

The object should have the following behaviors (methods) implemented:

- void addMerchandise(Merchandise merch)
  - The first addMerchandise call should set the Merchandise object of the Inventory, and update the count to 1.
  - Subsequent calls to addMerchandise behave as follows:
    - if the new item being added has the same SKU as the current Merchandise, update the available count
    - if the new item being added does not have the same SKU, do nothing
- boolean isAvailable(int SKU)
  - This method should return either true or false depending on whether there exists a merchandise item with such a SKU, and the number available is greater than 0.
- void buy(int SKU)
  - This method should remove a merchandise with a given SKU the available merchandise.
- void returnMerchandise(int SKU)
  - This method should add a merchandise with a given SKU to the list of available merchandise.
- boolean search(String title)
  - This method should return true or false depending on whether there exists a merchandise item with such a title.
- void printInventory()
  - This method should print to console a list of all the titles of all merchandise items and how many are available.
- void increasePrice(double priceIncrease)
  - This method should cycle through all the merchandise items in the Inventory and increase the price of each one by the amount specified by priceIncrease.

3 Suggestions & Submission Details

You may wish to implement a testing procedure to evaluate the correctness of your implementation. Your code will be tested using a test harness to ensure its correctness. In addition to submitting the .java files for all of the object implementations, please submit a brief write-up (pdf or plain-text files only) describing any difficulties you have encountered or any concerns you may have.