Project 7A: Apparel Item

Assigned: October 19, 2010
Due: Sunday, October 23, 2010 by 11:59 PM

Deliverables:

The following project files must be uploaded to Web-CAT by the due date and time specified above (see the Lab Guidelines for information on submitting project files). **If you are not able to submit your project via Web-CAT, then you must e-mail your project to gofflau@auburn.edu before the deadline. Therefore, it is important to give yourself time to submit via e-mail if you have trouble submitting to Web-CAT.**

- ApparelItem.java
- ApparelItemTest.java

Specifications:

ApparelItem.java

- **Requirements**: In this project you will create a class to represent a piece of apparel being sold in a store. Each item is represented by a code with the name, id (an int), and price separated by commas. For example an item with the code

  ```
  T Shirt,235364, 9.05
  ```

  would represent an item called "T Shirt" with id 235364 and price $9.05. There may or may not be spaces after the commas. Your program should keep track of the total sales made of a particular item, the total sales of all items, and the highest selling item.

- **Design**: Your ApparelItem class must have a constructor that takes the code as a single String (shown above, separated by commas) as a string. Your method must also have the following methods:
  - `setItemCode`: sets the item information given a String parameter for the code (such as "T Shirt, 235364, 9.05") and returns only true if there are exactly 3 items in the code. Returns false if the code is invalid (does not contain exactly 2 commas).
  - `getName`: returns the name of the item (no leading or trailing whitespace).
  - `getId`: returns the id of the item as an int.
  - `getPrice`: returns the price of the item as a double.
  - `sellItem`: Makes one sale of the item and adds the current price to the total sales for that particular item.
  - `totalItemSales`: returns the total sales for that item only (a double).
  - `allItemSales`: returns the total sales of all items (a double). Example: if an item is created with price $5 and two sales are made, and then an item is created with price $17 and one sales is made, then allItemSales should return $27. If no ApparelItem object has been instantiated, then the allItemSales method should return a value of 0.
  - `highestSeller`: returns an ApparelItem that represents the item with the current highest total item sales (highest totalItemSales of all times). Returns null until the first item is sold; must be invoked using the name of the class.

    ```
    ApparelItem.highestSeller()
    ```

  - `toString`: Your toString should return a String representation of the item including its name, its id, its price, and its total item sales.

- **Code**: You can use the split method of the String class to split the item code around the comma (use strObj.split(","), where strObj is the code). Also use the trim method to eliminate leading and trailing whitespace on each part of the code String. Be careful when determining which variables should be static and which should be instance variables.

- **Test**: Test your class using the ApparelItemTest file below. The tests do not fully cover the class functionality; add as many more as you would like.

ApparelItemTest.java

- **Test Requirements**: Download the ApparelItemTest file from the class website and add it to the same project as ApparelItem. In the project view, right-click ApparelItemTest and click Mark as Test. In order to receive full credit, you must complete each test according to the comments. If you add any methods of your own to ApparelItem then you will have to test those methods as well for full credit.