CSE 2010, HW1, Spring 2024

Due Thu Jan 25 at the start of your lab section; Submit Server: class = cse2010, assignment = hw1SxIndividual

Due Thu Jan 25 at the end of your lab section; Submit Server: class = cse2010, assignment = hw1SxGroupHelp

x is 14, 23—your section number.

On some shopping sites (e.g., amazon.com), a product can have multiple sellers. Usually, the seller list is displayed based on the total cost (product price plus shipping cost). Natually, a seller is listed if it has inventory for the product.

For this assignment, assume you are managing three products: appleIPhone, earBuds, and keyboard. To separately manage the seller list in *ascending* order of total cost for each product, use a singly linked list. Ties are broken by seller names in alphabetically order. When the inventory is zero, the corresponding node is deleted. You can borrow/modify SinglyLinkedList from the textbook or implement your own. We will evaluate your submissions on code01.fit.edu so we strongly recommend you to test your programs on code01.fit.edu. To preserve invisible characters, we strongly recommend you to download, NOT copy and paste, input data files.

Input: To simulate the shopping and inventory events, an input file contains events in the same directory as your program file called HW1.java that has the main method. Initially, each product has no sellers (no inventory). Your submission takes the input file name as a command-line argument. Each line is one of the following event:

- AddSeller *product seller price shippingCost quantity*
- RemoveSeller *product seller*
- IncreaseInventory *product seller quantity*
- CustomerPurchase *product seller quantity*
- DisplaySellerList *product*

DisplaySellerList considers the cost of one item of the product.

Output: The program prints events to the standard output (screen). Each event is on one line and possible events are:

- AddSeller *product seller price shippingCost quantity* [$NonPositiveQuantityError$]
- RemoveSeller *product seller* [$NonExistingSellerError$]
- IncreaseInventory *product seller quantity updatedInventory*
- CustomerPurchase *product seller quantity updatedInventory* or $NotEnoughInventoryError$
- DepletedInventoryRemoveSeller *product seller*
- DisplaySellerList *product*

```
12345678901234567890123456789012345678901234567890  // just to show spacing
    seller  productPrice  shippingCost  totalCost   // output starts
   walmart         20.99          0.00      20.99
    amazon         16.95          5.00      21.95
   bestbuy         21.99          0.00      21.99
```

Assume a seller's name is at most 10 characters, use right justification for each column in the seller list. When inventory is depleted, DepletedInventoryRemoveSeller is displayed and the corresponding node is removed.

Sample input and output files are on Canvas.

Submission: Submit HW1.java that has the main method and other program files (if any). Submissions from individual students are due at the beginning of their respective lab sections via assignment *hw1SxIndividual* (see the top).

During the lab session on the due date, we encourage students to bring test cases (beyond the sample input) to test and improve each other's program in the group. Improved programs are submitted via assignment *hw1SxGroupHelp*, which is due at the end of the lab session (see the top). Your program is mainly evaluated based on *hw1SxIndividual*. Improvement on test cases will receive half credit. Specifically, $testCaseImprovement(hw1) = testCaseScore(hw1SxGroupHelp) - testCaseScore(hw1SxIndividual)$; $testCaseScore(hw1) = testCaseScore(hw1SxIndividual) + testCaseImprovement(hw1)/2$. Note the late penalty on the syllabus if you submit after the due date and time as specified at the top.