Design/Coding Exercise

- You may not use any external libraries to solve the problem itself, but you may use external libraries or tools for building or testing purposes.

How we evaluate your code

We will be looking at a number of things including the design aspect of your solution and your object oriented programming skills. Whilst these are small problems, we expect you to submit what you believe is “production-quality” code that you would be able to run, maintain and evolve. You don’t need to “gold plate” your solution, but we are looking for something more than a bare-bones algorithm. You should submit code that you would be happy to produce in a real project, or that you would be happy to receive from a colleague. We recommend you to provide adequate tests to indicate full code coverage. We also recommend you follow good code hygiene and clean code practices to reduce cyclomatic complexity of your classes.

How to submit your code

Please upload the code to github, and submit a link.

Problem: Inventory Management 1

Mr. X owns a store that sells almost everything you think about. Now he wants a inventory management system to manage his inventory. Mr. X feels that controlling his inventory through SMS from his mobile will be revolutionary. So as a prequel, he decides that he wants a system that accepts one line commands and performs the respective operation.

Below is the list of commands he needs in the system:

a) create itemName costPrice sellingPrice

Whenever Mr. X wants to add a new item to his store he issues a create command. This command creates a new item in the inventory with the given cost price and selling price. The prices are rounded off to two decimal places.

b) delete itemName

If Mr. X decides not to sell an item anymore, then he simply issues a delete command. This command will remove the item from the inventory.

c) updateBuy itemName quantity

Whenever Mr. X purchases additional quantity of the mentioned item, then he issues a updateBuy command. This command should increase the quantity of the mentioned item.

d) updateSell itemName quantity

Whenever Mr. X sells some item, then he issues a updateSell command. This command should deduct the quantity of the mentioned item.

e) report

Whenever Mr. X wants to view his inventory list he issues the report command. This command should print the current inventory details in the specified format sorted by alphabetical order. Apart from printing the inventory it has to report on the profit made by Mr. X since last report generation.
Where profit is calculated by: $\sum$ (sellingPrice-costPrice) of the sold items multiplied by no. of items sold-costPrice of the deleted items.

Sample Input

```
create Book01 10.50 13.79
create Food01 1.47 3.98
create Med01 30.63 34.29
create Tab01 57.00 84.98
updateBuy Tab01 100
updateSell Tab01 2
updateBuy Food01 500
updateBuy Book01 100
updateBuy Med01 100
updateSell Food01 1
updateSell Food01 1
updateSell Tab01 2
report
delete Book01
updateSell Tab01 5
create Mobile01 10.51 44.56
updateBuy Mobile01 250
```

```
updateSell Food01 5
updateSell Mobile01 4
updateSell Med01 10
report
#
```

Expected Output

```
            INVENTORY REPORT
Item Name   Bought At         Sold At           AvailableQty      Value
---------   ---------         -------           ------------      -------
Book01      10.50             13.79                   100         1050.00
Food01      1.47              3.98                    498         732.06
Med01       30.63             34.29                   100         3063.00
Tab01       57.00             84.98                   96          5472.00
-------------------------------------------------------------------------
Total value                                                       10317.06
Profit since previous report                                            116.94


                  INVENTORY REPORT
Item Name   Bought At         Sold At     AvailableQty      Value
---------   ---------         -------     ------------      -------
Food01            1.47        3.98        493               724.71
Med01             30.63       34.29             90                2756.70
Mobile01          10.51       44.56             246               2585.46
Tab01             57.00       84.98             91                5187.00
-------------------------------------------------------------------------
Total value                                                       11253.87
Profit since previous report                                      -724.75
```

In-Office Extension:

f) updateSellPrice itemName newSellPrice

Mr. X is now happy with the current system, however at times some goods are not selling out faster than expected. In this case he wants to change the initial selling price of that item. So he wants a new command. This command will update the sellingPrice of the specified item.

report:

Now while calculating the report profit, the system should take care of 2 different selling prices for a same item. Profits should be calculated with old selling price for the items sold before updating and with new selling price for the items bought after updating.