Assignment 6 Instructions

1. Assignment 05: **40 points total with 2 E.C. points (For class participation, for extra work helping others in class, for not being late on submitting your assignment.)**
2. Due Date & Time: **10/19/2020 at 11:59 PM**

WHAT TO SUBMIT

Submit 2 files to iLearn by the deadline and post a reflection on iLearn. [38pts + 2 E.C. pts = 40 points]

- 1 Java File: Please submit 1 files to iLearn: **CoffeeShopAccountInventoryManager.java[30 points]**
- 1 PDF File: Submit 1 Word/PDF file which is a filled-out, downloaded local copy of this Google page on your local computer, named “firstname-lastname-assignment-6-report.pdf”. Fill this out with screenshots then save it as Word or PDF
- 1 Reflection with a Buddy on iLearn

HOW TO SUBMIT

Please upload all 2 files separately via iLearn Assignments Submission

GUIDELINES FOR ALL ASSIGNMENTS:

1. Each assignment includes a code portion and a non-code portion. Please submit both 2 portions.
   a. Code portion: Your source code files, only the files which you create and edit.
   b. Non-code portion: Your assignment report, only 1 **Word** or **PDF** file.

2. Please submit all required files separately, un-zipped, via iLearn Assignments Submission

3. Always **read through the entire assignment before starting and submitting any of it. Missing files or missing requirements will result in deducted points**
4. **a. Include a proper header at the top of every Java file. Figure 1**

Header Format
/*

```
* Assignment <assignment number>

* Description: <program description>

* Name: <your name>

* ID: <your SFSU ID number>

* Class: CSC 210-<section number>

* Semester: <current semester>

*/
```

Replace each tag (such as **<assignment number>**) with the appropriate text.

You should adhere to this format as closely as possible. You do not need to include the **<>** symbols in your header fields.

b. Only if you work with a Study Buddy, include your Buddy's name in your header at the top of every Java file. Figure 1

Header Format

```
/*

* Assignment <assignment number>

* Description: <program description>

* Name: <your name>

* Teammate: <Study Buddy name>

* ID: <your SFSU ID number>

* Class: CSC 210-<section number>

* Semester: <current semester>

*/
```

Assignment 5

Coffee Shop Accounting Operations

❐Part 1: CoffeeShopInventoryManager [10 points]

File Name: CoffeeShopInventoryManager.java

1. (Total 10 points) Declare and create 3 arrays.

a. (2 points) The first array, named items, should contain 5 items ("cups", "coffee beans", "lids", "towel", "sleeves"). Use Array-Initializer notation for assigning these 5 values to this array.

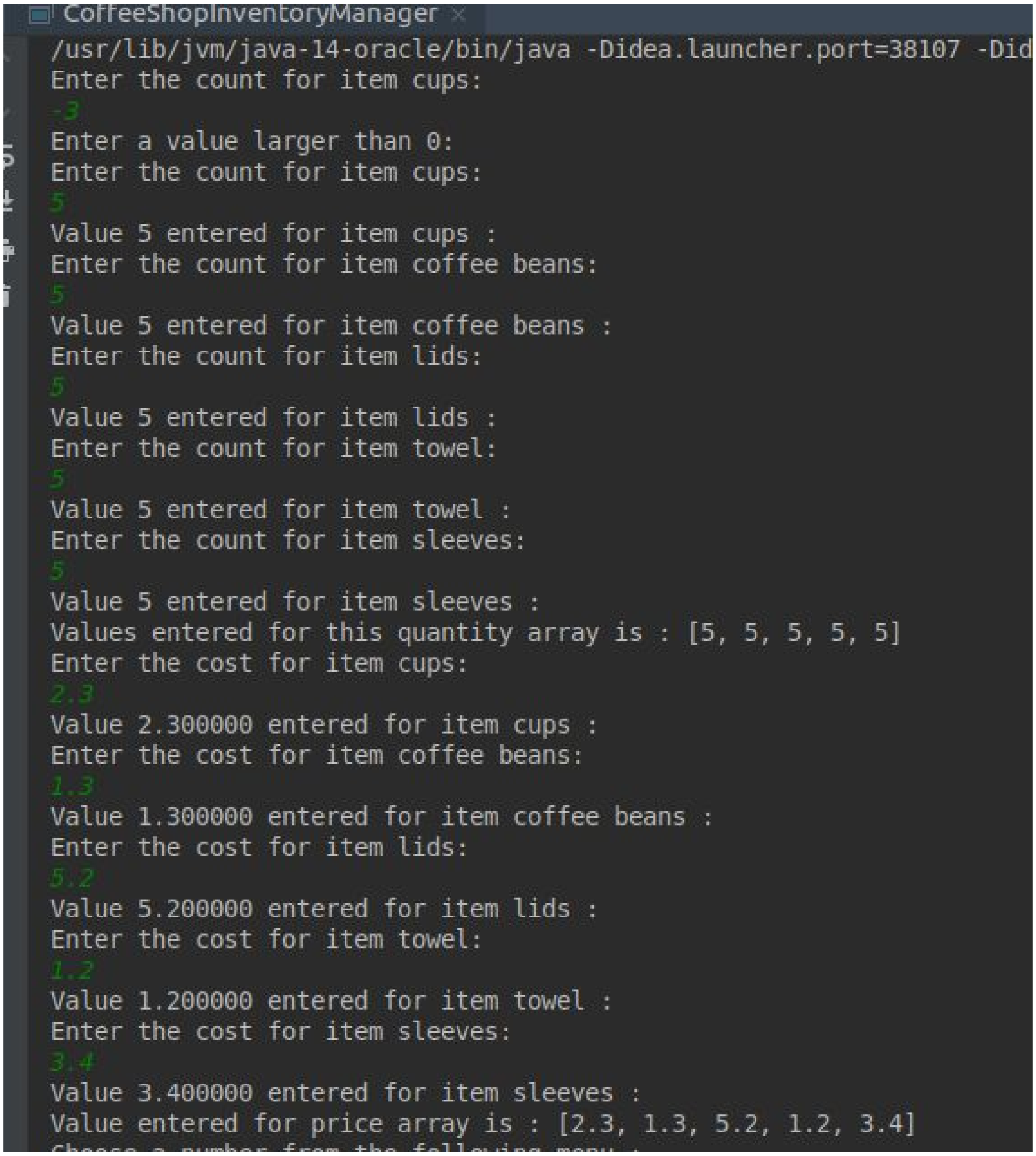
b. (3 points) The second array, named quantity, should contain the current quantity for each of the corresponding items. Obtain these quantities from the user. For each user input check if it is a valid number (greater than or equal to 0).

See output below:

c. (3 points) The third array, named price, should contain the price per piece of the corresponding item (example, 4 dollars for a cups, 0.5 dollars for a coffee beans). Obtain these prices from the user. For each user input check if it is a valid number (greater than or equal to 0).

See output below:

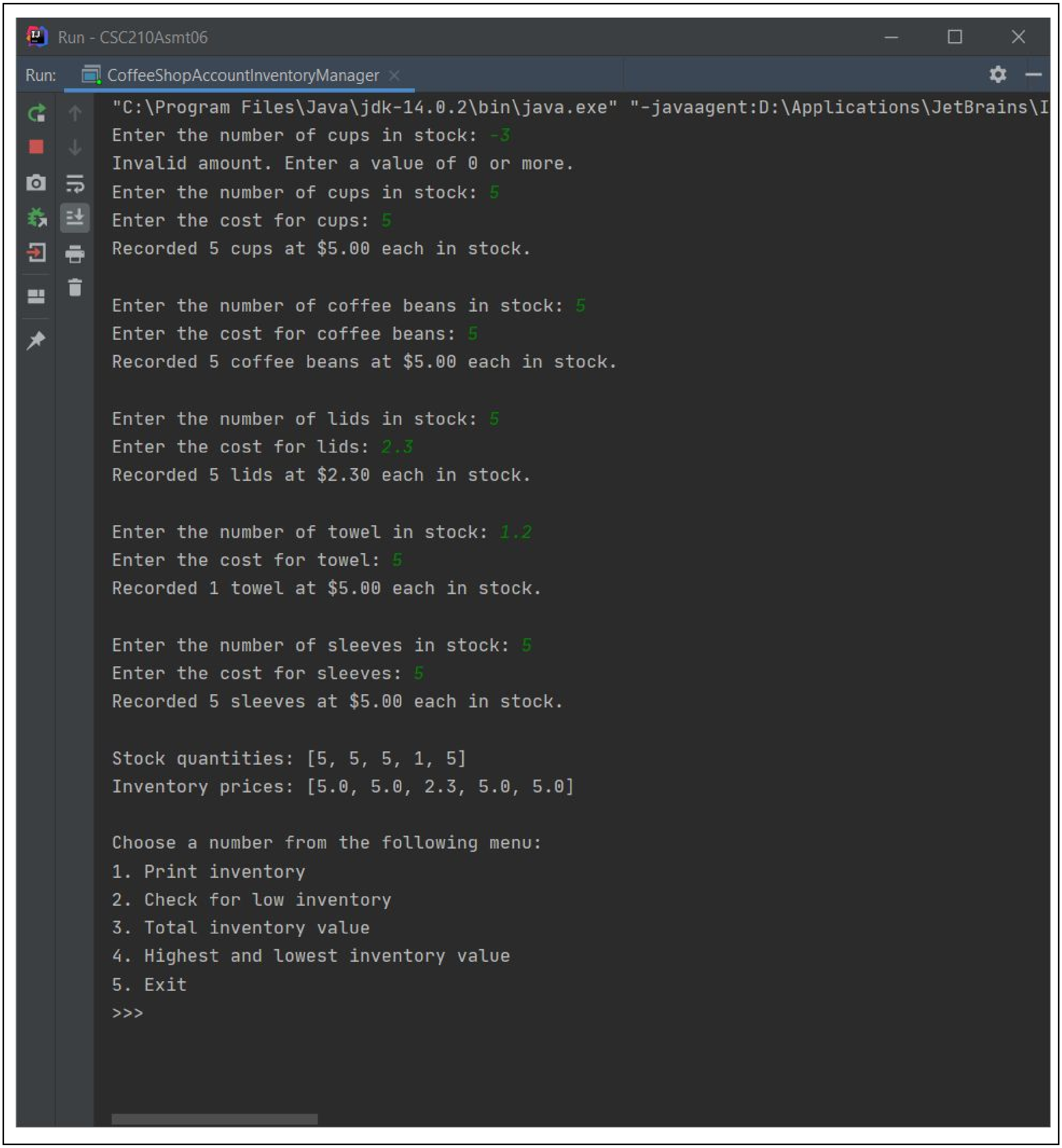
d. (2 points. 1 per array) Use the appropriate datatype for each array.



```
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.port=38107 -Did
Enter the count for item cups:
-3
Enter a value larger than 0:
Enter the count for item cups:
5
Value 5 entered for item cups :
Enter the count for item coffee beans:
5
Value 5 entered for item coffee beans :
Enter the count for item lids:
5
Value 5 entered for item lids :
Enter the count for item towel:
5
Value 5 entered for item towel :
Enter the count for item sleeves:
5
Value 5 entered for item sleeves :
Values entered for this quantity array is : [5, 5, 5, 5, 5]
Enter the cost for item cups:
2.3
Value 2.300000 entered for item cups :
Enter the cost for item coffee beans:
1.3
Value 1.300000 entered for item coffee beans :
Enter the cost for item lids:
5.2
Value 5.200000 entered for item lids :
Enter the cost for item towel:
1.2
Value 1.200000 entered for item towel :
Enter the cost for item sleeves:
3.4
Value 3.400000 entered for item sleeves :
Value entered for price array is : [2.3, 1.3, 5.2, 1.2, 3.4]
Choose a number from the following menu :
```

Please paste your screenshot for this scenario here:

Output :

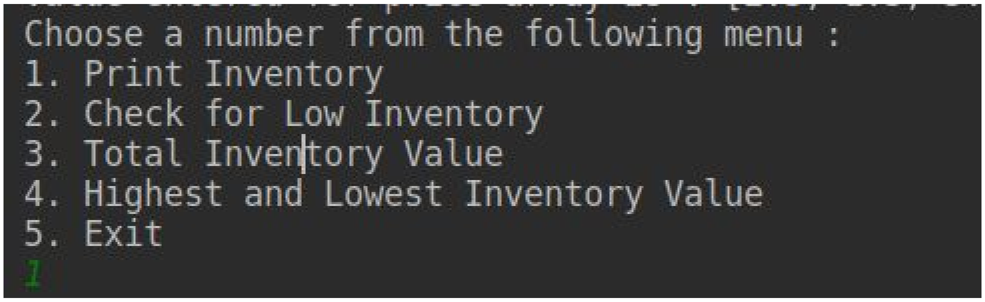


```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBrains\I
Enter the number of cups in stock: -3
Invalid amount. Enter a value of 0 or more.
Enter the number of cups in stock: 5
Enter the cost for cups: 5
Recorded 5 cups at $5.00 each in stock.

Enter the number of coffee beans in stock: 5
Enter the cost for coffee beans: 5
Recorded 5 coffee beans at $5.00 each in stock.

Enter the number of lids in stock: 5
Enter the cost for lids: 2.3
Recorded 5 lids at $2.30 each in stock.

Enter the number of towel in stock: 1.2
Enter the cost for towel: 5
Recorded 1 towel at $5.00 each in stock.

Enter the number of sleeves in stock: 5
Enter the cost for sleeves: 5
Recorded 5 sleeves at $5.00 each in stock.

Stock quantities: [5, 5, 5, 1, 5]
Inventory prices: [5.0, 5.0, 2.3, 5.0, 5.0]

Choose a number from the following menu:
1. Print inventory
2. Check for low inventory
3. Total inventory value
4. Highest and lowest inventory value
5. Exit
>>>
```

❒Part 2: CoffeeShopInventoryManager [20 points]

2. (Total 20 points) A typical user of this program is the inventory manager.

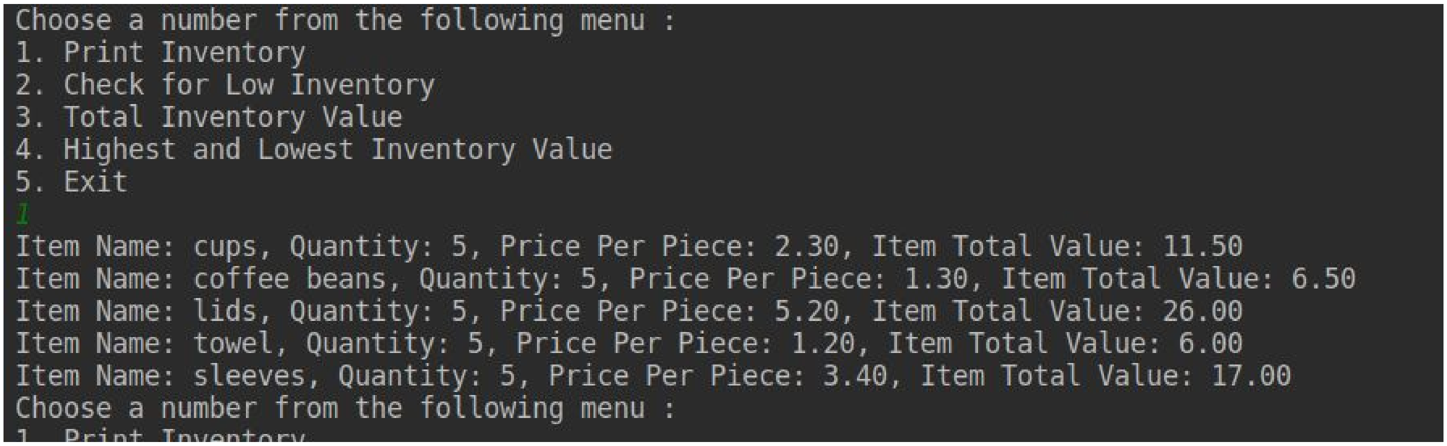
a. (3 points) Display to the user the possible operations on the inventory and prompt her/him to choose one. There are 5 possible operations: Print Inventory, Check for low inventory, Total inventory value, Highest and lowest inventory value items, Exit. The details of each of these operations are given below.



```
Choose a number from the following menu :
1. Print Inventory
2. Check for Low Inventory
3. Total Inventory Value
4. Highest and Lowest Inventory Value
5. Exit
1
```

b. (3 points) Print inventory (p): This operation prints the inventory in the following format:

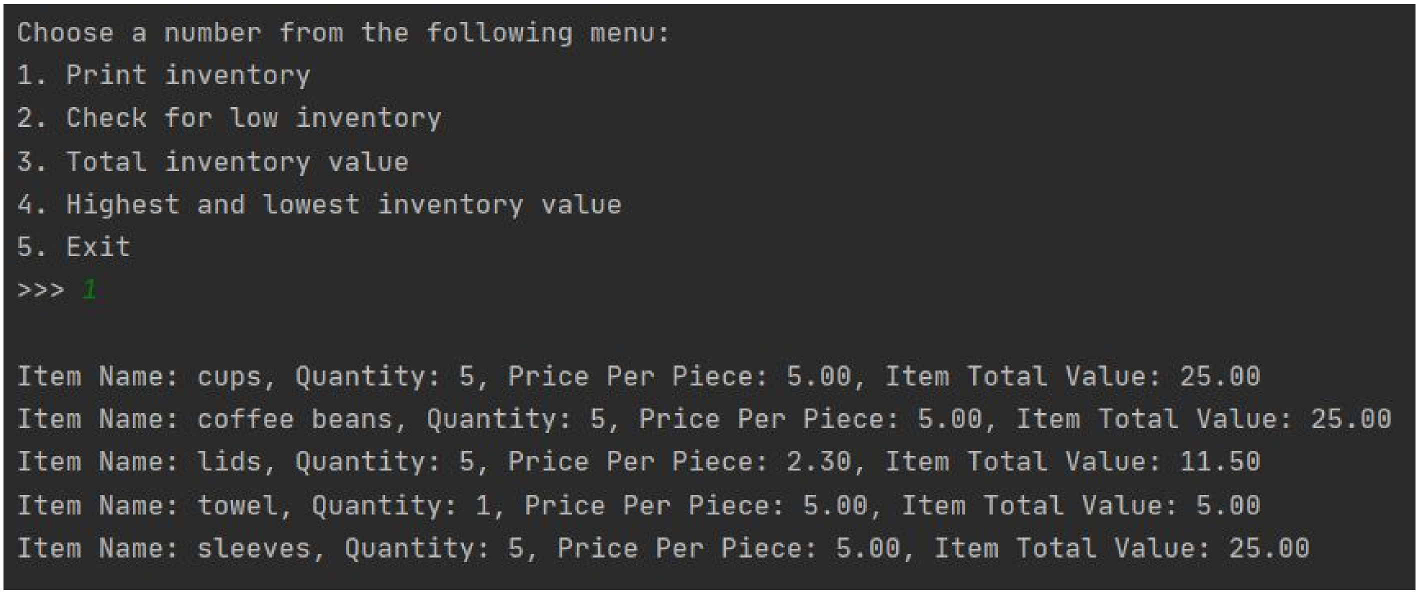
Item Name, Quantity, Price Per Piece, Item Total Value. The last column prints the product of quantity and price per piece for each item. Go back to step a.



```
Choose a number from the following menu :
1. Print Inventory
2. Check for Low Inventory
3. Total Inventory Value
4. Highest and Lowest Inventory Value
5. Exit
1
Item Name: cups, Quantity: 5, Price Per Piece: 2.30, Item Total Value: 11.50
Item Name: coffee beans, Quantity: 5, Price Per Piece: 1.30, Item Total Value: 6.50
Item Name: lids, Quantity: 5, Price Per Piece: 5.20, Item Total Value: 26.00
Item Name: towel, Quantity: 5, Price Per Piece: 1.20, Item Total Value: 6.00
Item Name: sleeves, Quantity: 5, Price Per Piece: 3.40, Item Total Value: 17.00
Choose a number from the following menu :
1. Print Inventory
```

Please paste your screenshot for this scenario here:

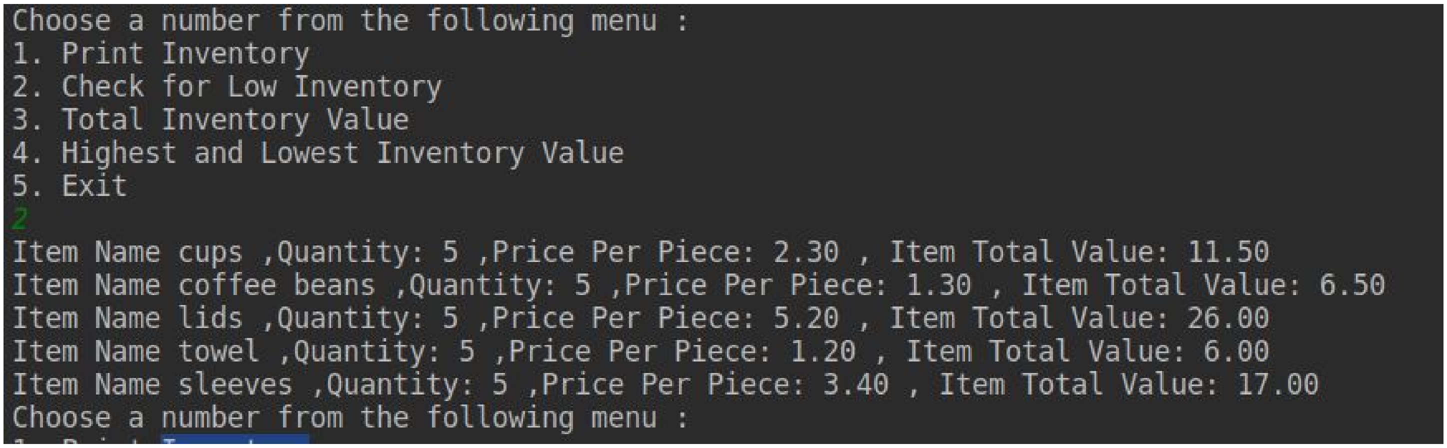
Output :



```
Choose a number from the following menu:
1. Print inventory
2. Check for low inventory
3. Total inventory value
4. Highest and lowest inventory value
5. Exit
>>> 1

Item Name: cups, Quantity: 5, Price Per Piece: 5.00, Item Total Value: 25.00
Item Name: coffee beans, Quantity: 5, Price Per Piece: 5.00, Item Total Value: 25.00
Item Name: lids, Quantity: 5, Price Per Piece: 2.30, Item Total Value: 11.50
Item Name: towel, Quantity: 1, Price Per Piece: 5.00, Item Total Value: 5.00
Item Name: sleeves, Quantity: 5, Price Per Piece: 5.00, Item Total Value: 25.00
```

c. (5 points) Check for low inventory (c): This operation checks for items that have 5 or fewer quantity, and prints them in the same format as in option(b).

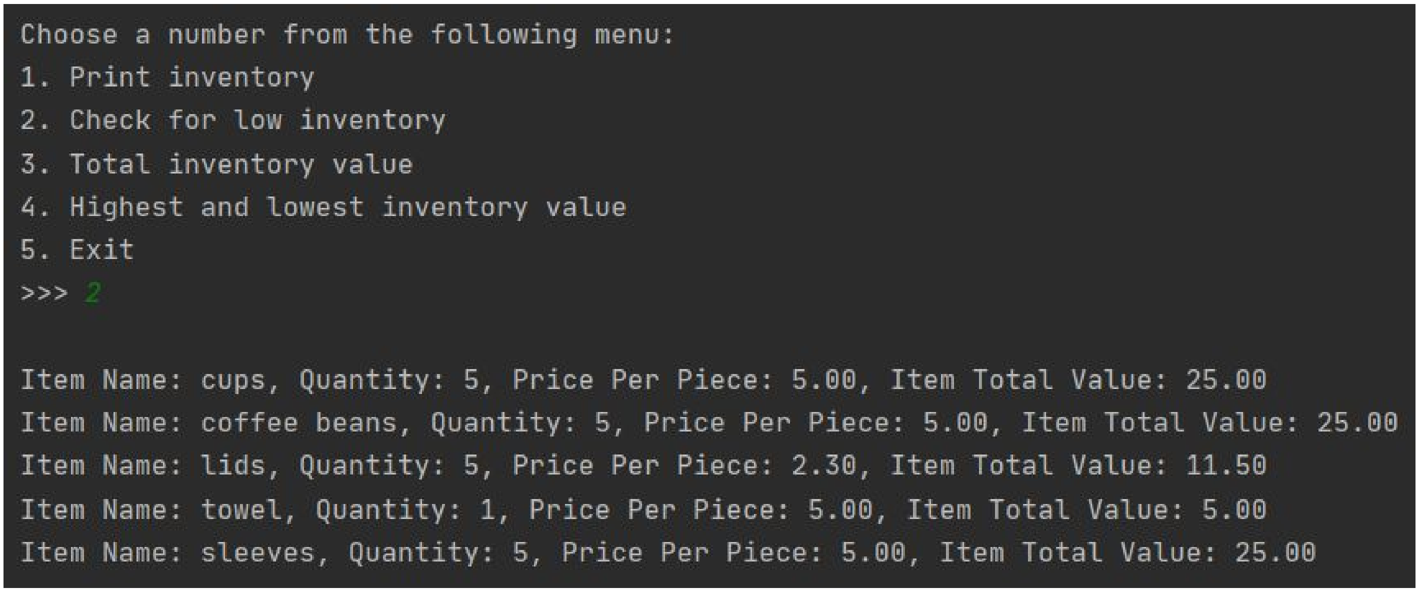
If there is no such item then print an appropriate message. Finally, go back to step a.



```
Choose a number from the following menu :
1. Print Inventory
2. Check for Low Inventory
3. Total Inventory Value
4. Highest and Lowest Inventory Value
5. Exit
2
Item Name cups ,Quantity: 5 ,Price Per Piece: 2.30 , Item Total Value: 11.50
Item Name coffee beans ,Quantity: 5 ,Price Per Piece: 1.30 , Item Total Value: 6.50
Item Name lids ,Quantity: 5 ,Price Per Piece: 5.20 , Item Total Value: 26.00
Item Name towel ,Quantity: 5 ,Price Per Piece: 1.20 , Item Total Value: 6.00
Item Name sleeves ,Quantity: 5 ,Price Per Piece: 3.40 , Item Total Value: 17.00
Choose a number from the following menu :
```

Please paste your screenshot for this scenario here:

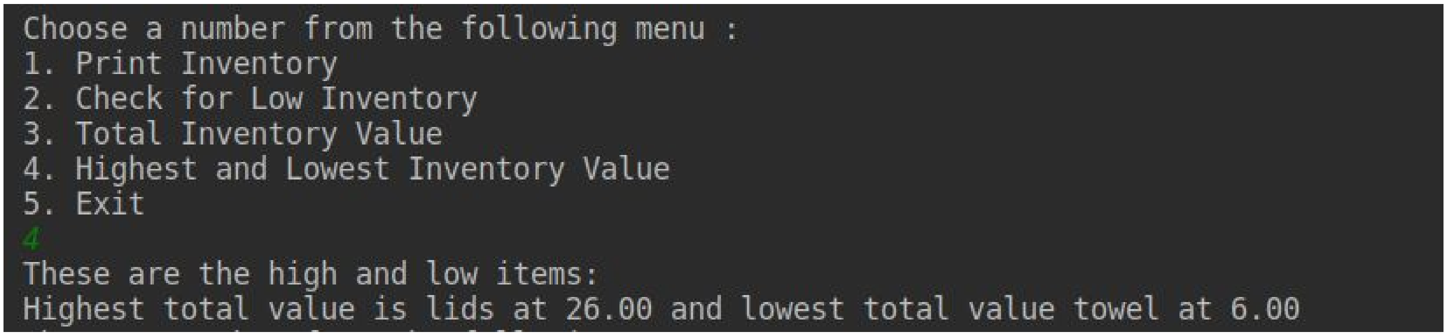
Output :



```
Choose a number from the following menu:
1. Print inventory
2. Check for low inventory
3. Total inventory value
4. Highest and lowest inventory value
5. Exit
>>> 2

Item Name: cups, Quantity: 5, Price Per Piece: 5.00, Item Total Value: 25.00
Item Name: coffee beans, Quantity: 5, Price Per Piece: 5.00, Item Total Value: 25.00
Item Name: lids, Quantity: 5, Price Per Piece: 2.30, Item Total Value: 11.50
Item Name: towel, Quantity: 1, Price Per Piece: 5.00, Item Total Value: 5.00
Item Name: sleeves, Quantity: 5, Price Per Piece: 5.00, Item Total Value: 25.00
```

d. (5 points) Highest and lowest inventory value items (h): This operation finds the item with the highest inventory value (quantity*price-per-piece),

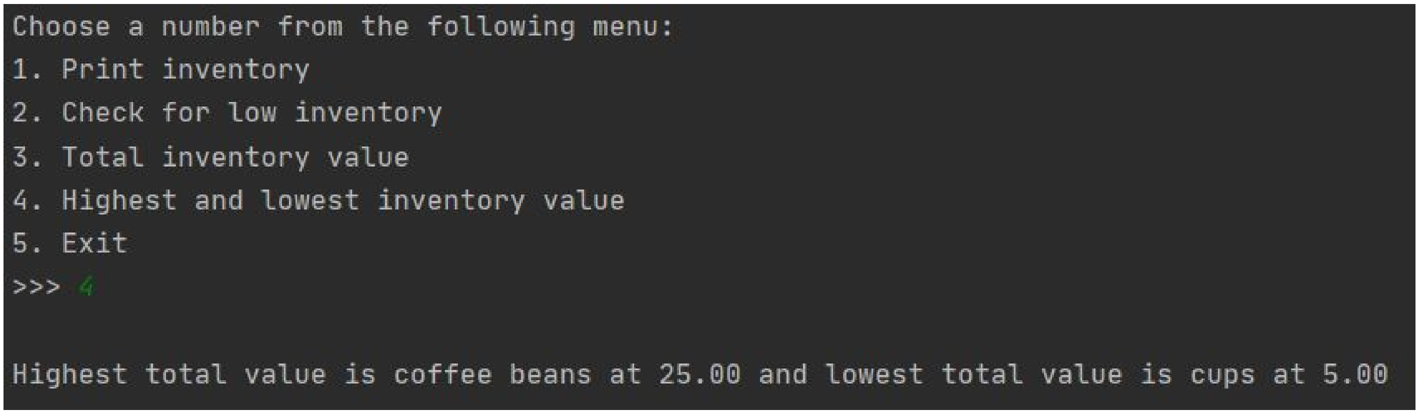
If there is more than one item with the same highest (or lowest) value then display all such items. Finally, go back to step a.



```
Choose a number from the following menu :
1. Print Inventory
2. Check for Low Inventory
3. Total Inventory Value
4. Highest and Lowest Inventory Value
5. Exit
4
These are the high and low items:
Highest total value is lids at 26.00 and lowest total value towel at 6.00
```

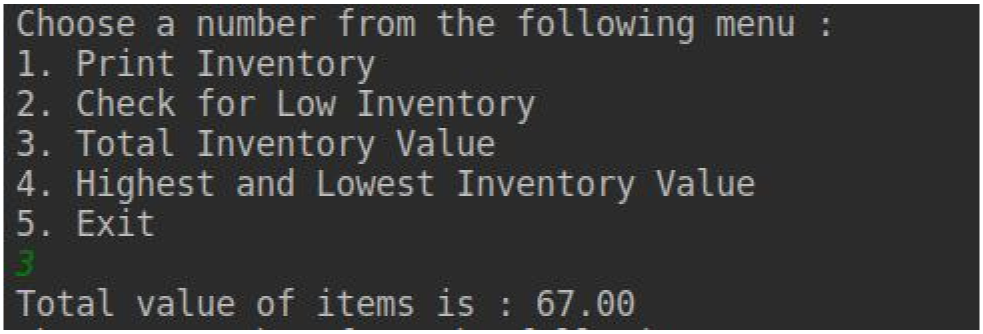
Please paste your screenshot for this scenario here:

Output :



```
Choose a number from the following menu:
1. Print inventory
2. Check for low inventory
3. Total inventory value
4. Highest and lowest inventory value
5. Exit
>>> 4

Highest total value is coffee beans at 25.00 and lowest total value is cups at 5.00
```

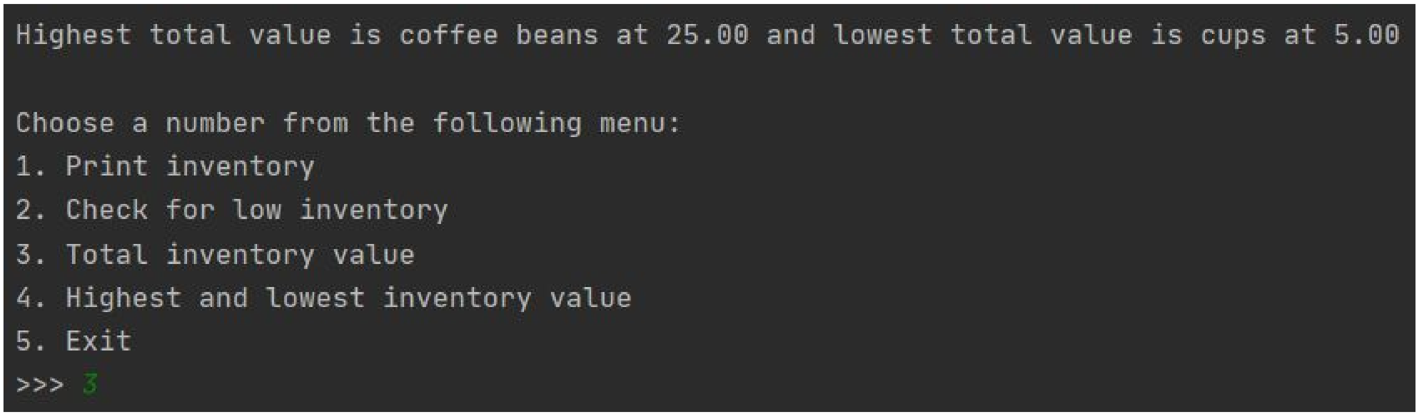
e. (3 points) This operation computes the grand total value of the current inventory using the quantity and price per piece information, and prints that grand total value. Finally, go back to step a.



```
Choose a number from the following menu :
1. Print Inventory
2. Check for Low Inventory
3. Total Inventory Value
4. Highest and Lowest Inventory Value
5. Exit
3
Total value of items is : 67.00
```

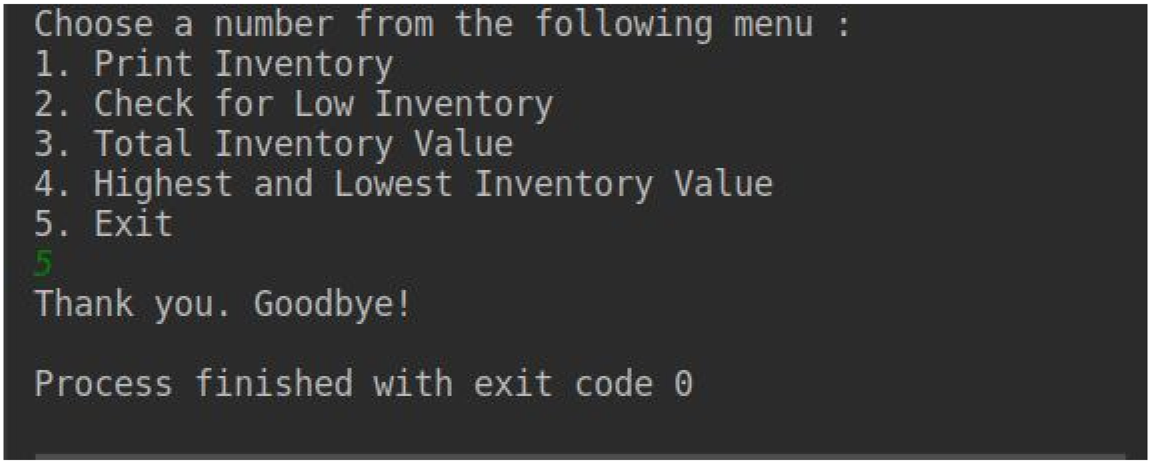
Please paste your screenshot for this scenario here:

Output :



```
Highest total value is coffee beans at 25.00 and lowest total value is cups at 5.00

Choose a number from the following menu:
1. Print inventory
2. Check for low inventory
3. Total inventory value
4. Highest and lowest inventory value
5. Exit
>>> 3
```

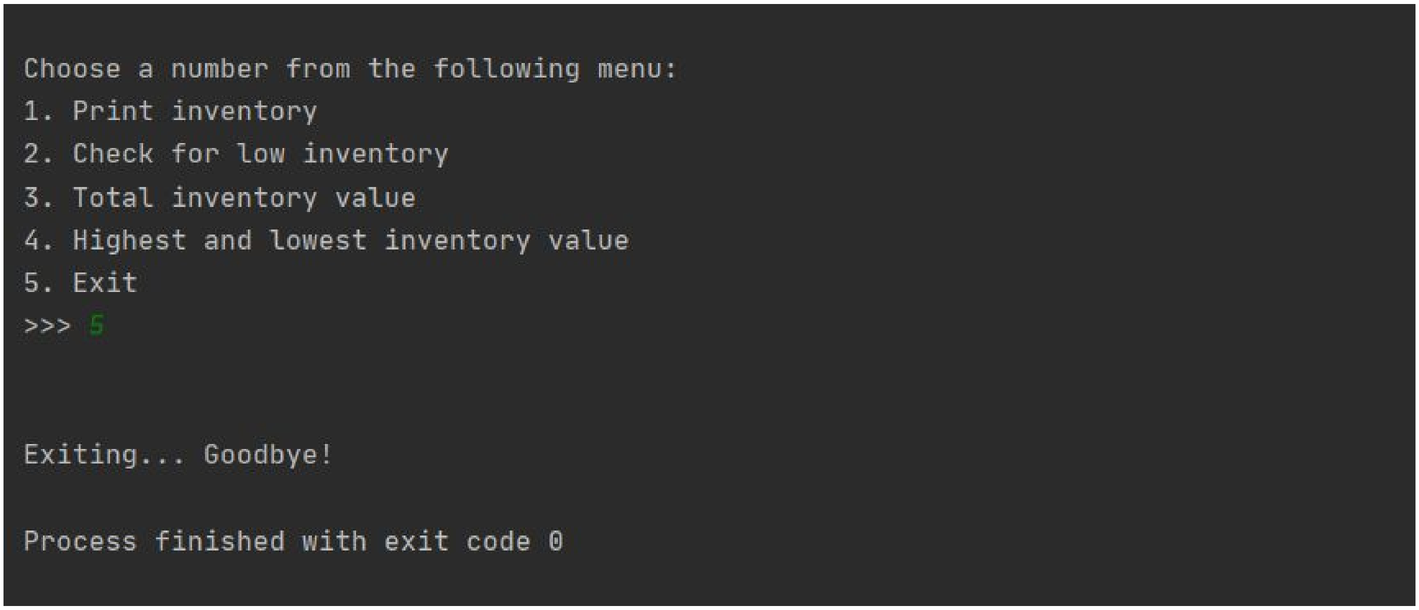
f. (1 point) Exit (e): Exits the program.



```
Choose a number from the following menu :
1. Print Inventory
2. Check for Low Inventory
3. Total Inventory Value
4. Highest and Lowest Inventory Value
5. Exit
5
Thank you. Goodbye!

Process finished with exit code 0
```

Please paste your screenshot for this scenario here:

Output :



```
Choose a number from the following menu:
1. Print inventory
2. Check for low inventory
3. Total inventory value
4. Highest and lowest inventory value
5. Exit
>>> 5


Exiting... Goodbye!

Process finished with exit code 0
```

SUBMISSION INSTRUCTIONS

1. Submit the 1 **CoffeeShopInventoryManager.java** file directly on iLearn

❐Part 3: Reflect with a Buddy 50 words [8 points]

1. Find your buddy. Look at the Study Buddy sheet and pair up with your assigned buddy. Or find someone to partner this task with.

2. You can set up a Slack, Discord, or a Zoom with your buddy to asynchronously or synchronously talk about anything, and also reflect on what was helpful and not helpful in completing this homework.

3. Each student will need to create 1 post to mark this assignment as complete.

Have fun with this one!!

👋 Assignment 6 Part 3: Reflection with a Buddy