

CSE 101 Programming Assignment 2

Page 1 of 4

Due:

20-November-2019 by 23:59

Deliverables:

The following Java file should be submitted to Google Classroom by the due date and time specified above. Submissions received after the deadline will be subject to the late policy described in the syllabus.

- Store_{StudentNumber}.java

Specifications:

Overview: You will continue the program this week to maintain the inventory for a store. Do not forget your headers with @author and @since information. This program will be expanded in future weeks, so be sure you understand the concepts covered in this program.

Requirements: Write a program that will simulate the user interface for a store. It will first prompt the store owner for four (4) products where each will have:

1. Name – String
 - a. Should be assigned as the value the owner enters with the first letter capitalized and all other letters lower case
2. Quantity – Integer
 - a. If the value passed is negative, the assigned quantity should be the positive value
3. Price – Floating point number
 - a. If the value passed is negative, the assigned price should be the positive value

Once the owner has entered the four products' information, the program will then prompt the customer for which of the four products he/she would like to purchase. Once the customer has completed the order, the program will display their total.

Finally, the program will ask for the amount the customer gives to pay for his/her items.

To facilitate the execution of this program, you will write (at minimum) the following methods:

1. main()
 - a. A method to run the Store program
 - b. It will ask the store owner for the four product information.
 - c. It will then run according to the description above until the customer types "0" for the prompt.
 - d. Returns None
2. capitalize(name)
 - a. Format name with the first letter uppercase; all other letters lowercase
 - b. Takes one String type parameter
 - c. Returns a String

3. `validRequest(quantity, request)`
 - a. A method to verify the requested amount is valid
 - b. Takes two integer type parameters for the current quantity and the number requested by the customer
 - c. Returns a boolean
 - i. True if request is positive and less than or equal to quantity
 - ii. False for all other values
4. `cashReturned(amount)`
 - a. A method to determine the type of bills and coins to be given
 - b. Takes one parameter for the amount to be given
 - c. Returns a String containing the number of each type of bill in the format of "# - Type" (i.e. "2 - 20", "3 - 1"), each on a separate line.
 - i. The values tested will be standard TL types (200, 100, 50, 20, 10, 5, 1, 0.50, 0.25, 0.10, 0.05, 0.01)
5. Any other methods you feel helpful can be implemented, however, these will be the only methods tested.

Design:

When `capitalize` is called with the given parameters, the following values should be returned:

```
Store_123456789.capitalize("TEST")
Test
Store_123456789.capitalize("anotherTest")
Anothertest
```

When `validRequest` is called with the given parameters, the following values should be returned:

```
Store_123456789.validRequest(5, 2)
true
Store_123456789.validRequest(5, 6)
false
Store_123456789.validRequest(5, -2)
false
```

When the method `cashReturned` is called with the following values, the returned String should be one of these shown.

```
Store_123456789.cashReturned(386.86)
1 - 200
1 - 100
1 - 50
1 - 20
1 - 10
1 - 5
1 - 1
1 - 0.50
1 - 0.25
1 - 0.10
1 - 0.01
Store_123456789.cashReturned(290)
1 - 200
1 - 50
2 - 20
```

```
Store_123456789.cashReturned(386.86)
1 - 200
1 - 100
1 - 50
1 - 20
1 - 10
1 - 5
1 - 1
1 - 0.50
1 - 0.25
1 - 0.10
0 - 0.05
1 - 0.01
Store_123456789.cashReturned(290)
1 - 200
0 - 100
1 - 50
2 - 20
0 - 10
0 - 5
0 - 1
0 - 0.50
0 - 0.25
0 - 0.10
0 - 0.05
0 - 0.01
```

When run, your program should ask the owner for the product information, then the customer for what he/she would like to purchase. When the customer decides to checkout, the total will be shown and the amount given will be entered. If the amount is enough, the change will be given and the remaining amounts will be displayed.

```
***** Store Inventory Entry *****
Enter the name of product 1: bread
Enter the number of Bread we have: -10
Enter the cost of Bread: 0.75
Enter the name of product 2: COLA
Enter the number of Cola we have: 20
Enter the cost of Cola: -2

***** Customer Entry *****
Welcome to our store, we have the following. Please enter what you would like:
1 - Bread
2 - Cola
0 - to checkout
2
How many Cola would you like? 2
Please enter what you would like:
1 - Bread
2 - Cola
0 - to checkout
1
How many Bread would you like? 3
Please enter what you would like:
1 - Bread
2 - Cola
0 - to checkout
0

Please enter what you would like:
1 - Bread
2 - Cola
0 - to checkout
0

***** Customer Total *****
Bread - 3 * 0.75 = 2.25
Cola - 2 * 2.0 = 4.0
-----
Total due - 6.25

Please enter amount given
10
Thank you for your business. Your change given is:
3 - 1
1 - 0.50
1 - 0.25

***** Remaining Amounts *****
Bread - 7
Cola - 18
```

If the customer enters an invalid choice or invalid amount requested, the program should display an error message. Also, the program should keep asking for an amount given until a total that is greater than or equal to the total is entered. Examples shown below:

How many Bread would you like? 45 ERROR: Invalid request Please enter what you would like: 1 - Bread 2 - Cola 0 - to checkout	***** Customer Total ***** Bread - 5 * 0.75 = 3.75 Cola - 2 * 2.0 = 4.0 ----- Total due - 7.75 Please enter amount given 5 Not enough payment given Please enter amount given
----------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Code: Create variables for the various values entered by the user and assign it using the Scanner object. As a user enters valid values update the values of the variables accordingly.

Test: You are responsible for testing your program. It is important to not rely solely on the examples presented in this Project description.

Grading:

Google Classroom Submission: If anything is ambiguous, it is your responsibility to ask questions. It is also your responsibility to complete this assignment in a timely manner. E-mails with questions regarding this assignment will likely not be answered if received after 17:00 on the due date of the assignment. You can submit multiple times using Google Classroom, however, we will only grade the last version that you submitted.

Filename: You must name your java file according to the description above. If your file is not named in this way, your submission for this assignment will not be accepted.

Quiz in Lab: There will be a quiz based on this assignment given on 22-November. The result of this quiz will be used to determine your grade on this assignment. **Note:** if you do not attend the lab and take the quiz, your score on this assignment **will be 0.**