CSE 101 Programming Assignment 3

Due:

9-December-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 take arrays as parameters for the starting inventory. Then will ask the customer to request items, show a total, take a payment from the customer and display the change total.

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

1. displayMenu(item, price)
    a. A new method to display the options based on the contents of the arrays
    b. Takes an array of strings and an array of decimal numbers as parameters
    c. Displays to the output each of the name and price of each item in the arrays preceded by a consecutive number ending with "0 - to checkout"
    d. Returns None

```
Store_123456789.displayMenu(new String[]{"Item 1", "Item 2"}, new double[]{1, 2});
1 - for Item 1 (1.0)
2 - for Item 2 (2.0)
0 - to checkout
```

2. store(item, quantity, price)
    a. A method to run the store program
    b. It will take three parameters
        i. String array for names of the items in stock
            1. For each item in this array it will change the name to capitalize the first letter and lowercase other letters
        ii. Integer array for number of each item
        iii. Decimal array for price of each item
    c. It will run according to the description above until the user types "0" for the prompt.
    d. NOTE: This method will use much of the main method from your previous assignment
    e. Returns None

3. capitalize(name)
    a. For each value in the name array, format name with the first letter uppercase; all other letters lowercase
    b. Takes an array of Strings as a parameter
    c. Returns none
    d. **Note:** You can call your previous capitalize method from this method using method overloading

```
String[] arr = {"ITEM 1", "item 2"};
Store_123456789.capitalize(arr);
arr[0]
Item 1
arr[1]
Item 2
```

4. validRequest(quantity, request)
    a. No change
5. cashReturned(amount)
    a. No change
6. Any other methods you feel helpful can be implemented, however, these will be the only methods tested.

Design:
When store is called, your program should display the list of items and prices to the customer and ask 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. The customer should be able to add more items of a given type (i.e. the customer may decide to buy more of something).

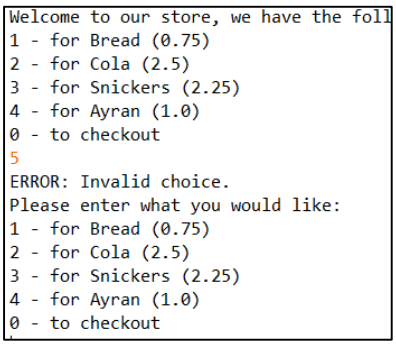
The example on the next page was executed using the following code:

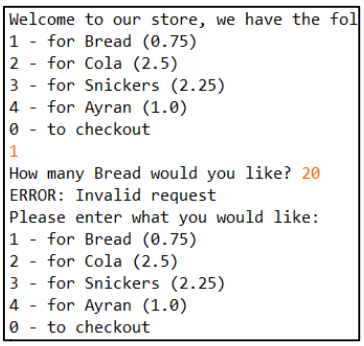
```
public static void main(String[] args) {
   String[] item = {"bread", "cola", "snickers", "AYRAN"};
   int[] quantity = {10, 15, 12, 30};
   double[] price = {.75, 2.5, 2.25, 1};
   store(item, quantity, price);
}
```

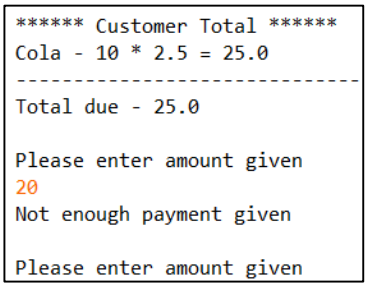
Note: The code above was a main method used to execute the store program. You are not required to have a main method. You are encouraged to use a main method to test your store program. If you include a main method, it will not be used for grading purposes.

```
Welcome to our store, we have the following. Please enter what you would like:
1 - for Bread (0.75)
2 - for Cola (2.5)
3 - for Snickers (2.25)
4 - for Ayran (1.0)
0 - to checkout
1
How many Bread would you like? 2
Please enter what you would like:
1 - for Bread (0.75)
2 - for Cola (2.5)
3 - for Snickers (2.25)
4 - for Ayran (1.0)
0 - to checkout
2
How many Cola would you like? 2
Please enter what you would like:
1 - for Bread (0.75)
2 - for Cola (2.5)
3 - for Snickers (2.25)
4 - for Ayran (1.0)
0 - to checkout
1
How many Bread would you like? 1
Please enter what you would like:
1 - for Bread (0.75)
2 - for Cola (2.5)
3 - for Snickers (2.25)
4 - for Ayran (1.0)
0 - to checkout
0

****** Customer Total ******
Bread - 3 * 0.75 = 2.25
Cola - 2 * 2.5 = 5.0
------------------------------
Total due - 7.25

Please enter amount given
10
Thank you for your business. Your change given is:
2 - 1
1 - 0.50
1 - 0.25
```

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:







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 13-December. 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**.