**Project 01: Shop Program**
**CSE-021-01, Spr 2018** **Points: 100**

## Overview

Create a new class called Shop (`Shop.java`) to do all your work. You may work in pairs or by yourself. The expectation is that you work on the project outside of lab time but you may certainly use any extra time in lab this week. Your partner can be in any section, both of you will make your own submission while noting your collaborator/partner in the text box of the submission.

## Project Requirements

You have been hired by a consultant firm to work on a shop program (`Shop.java`). It will have a very simple interface with the following 4 options:

- Setup shop
- Buy items
- List of items purchased
- Checkout

The idea is that the user should follow the sequence of setting up shop then letting the customer purchase items and check out. Each of the options should utilize at least one method each (4 methods minimum for the project). You may use more as you find necessary but this project should illustrate competence in method usage. The output for the `intro` method is as follows:

```
This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want:
```

### I. Setup Shop

1) Ask for the number of items to sell
2) For each item
   a) Ask the name of the item (one word)
   b) Ask the price of the item
3) Discount
   a) Ask for the threshold (over which amount to give discount)
   b) Ask for the rate (how much % discount)
4) User can run setup multiple times so keep the latest version

### II. Buy Items

1) If setup is not yet done, then ask user to setup shop first
2) For each item
   a) Ask the amount they wish to purchase
3) Customer can purchase multiple times, so only process the latest order (think of `init()` from labs)

### III. List of Items

1) If setup is not yet done, then ask user to setup shop first
2) If setup is done but the customer hasn't bought anything, then ask customer to buy first
3) For each item purchased (non-zero amount)
   a) Display amount purchased and price per item

**IV. Check out**

1) If setup is not yet done, then ask user to setup shop first
2) If setup is done but the customer hasn't bought anything, then ask customer to buy first
3) Display the summary
   a) Sub total
   b) Discount
   c) Total
4) End the program

## Sample Behavior

Your job is to **fool-proof** your program. If the customer does not follow the ascribed order then your program should provide the customer guidance on what to do first. Here is a sample of this behavior:

```
This program supports 4 functions:
         1. Setup Shop
         2. Buy
         3. List Items
         4. Checkout
Please choose the function you want: 2

Shop is not setup yet!

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 4

Shop is not setup yet!

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want:
```

The program will keep asking the 4 main questions until the customer decides to setup Shop. Here is a sample run with one misstep by the customer:

```
This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 1

Please enter the number of items: 2
Enter name of product 0: Shampoo
Enter price of product 0: 13.13
Enter name of product 1: Conditioner
Enter price of product 1: 10.99
Please enter the amount to qualify for discount: 100
Please enter the discount rate (0.1 for 10%): .05

This program supports 4 functions:
   1. Setup Shop
```

```
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 3

Try again: You have not bought anything

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 2

Enter the amount of Shampoo: 10
Enter the amount of Conditioner: 20

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 3

10 count of Shampoo @ 13.13 = $131.3
20 count of Conditioner @ 10.99 = $219.8

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 4

Thanks for coming!
Sub Total: $351.1
-Discount: $17.555000000000003
Total:     $333.545
```

And the following is a sample run when a customer purchases multiple times with different amounts:

```
This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 1

Please enter the number of items: 3
Enter name of product 0: Shampoo
Enter price of product 0: 10.99
Enter name of product 1: Conditioner
Enter price of product 1: 6.99
Enter name of product 2: Combo
Enter price of product 2: 12.99
Please enter the amount to qualify for discount: 50
Please enter the discount rate (0.1 for 10%): .1

This program supports 4 functions:
```

```
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 2

Enter the amount of Shampoo: 1
Enter the amount of Conditioner: 1
Enter the amount of Combo: 1

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 3

1 count of Shampoo @ 10.99 = $10.99
1 count of Conditioner @ 6.99 = $6.99
1 count of Combo @ 12.99 = $12.99

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 2
Enter the amount of Shampoo: 0
Enter the amount of Conditioner: 1
Enter the amount of Combo: 0

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 3

1 count of Conditioner @ 6.99 = $6.99

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 2

Enter the amount of Shampoo: 1
Enter the amount of Conditioner: 0
Enter the amount of Combo: 1

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 3

1 count of Shampoo @ 10.99 = $10.99
1 count of Combo @ 12.99 = $12.99
```

```
This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 4

Thanks for coming!
Sub Total: $23.98
-Discount: $0.0
Total:     $23.98
```

The user might enter inputs other than 1, 2, 3, or 4. Your program must catch such invalid inputs as shown below:

```
This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want: 5

Error: Do not know 5

This program supports 4 functions:
   1. Setup Shop
   2. Buy
   3. List Items
   4. Checkout
Please choose the function you want:
```

## Test Cases and Console Log

The requirements for this project are listed at the beginning of this document labelled I – IV, with sub items numbered 1 – 4. You will need to provide test cases for each requirement as part of your submission. An example of a few test cases are as follows:

| **Requirement** | **Test case** |
|---|---|
| I – 1: | Created quantities of 1, 2 and 10 to sell |
| II – 1: | Input 2 before 1 and check to ensure program outputs appropriate message and presents the 4 functions to the user again. |
| II – 3: | Ordered 5 of item 1 first time but did not purchase when ordering again, checked the output items to make sure it's 0 instead of 5 (no corruption of the orders) |
| III – 1: | Input 3 before 1 |
| III – 2: | Input 3 before 2 |

You will need this for EVERY requirement and how you tested for each portion of the project. NOTE: the example above is very incomplete. You may also choose to group multiple requirements with a single test case. In this case you should clearly indicate all the requirements covered by the test case. List your test cases in a file called **`Test_cases.txt/doc/docx`**. This should purely be a list of all the different test cases you have created yourself, and used to verify your program is correct.

You will also include a log (copy and paste from the Console output) into a file called **`Test_cases.log`** -- this is simply a text document that you rename to Test_cases.log. Note that you must make sure to have the file name extensions visible in order to rename the file properly. The log file is merely all the tests you listed in the txt/doc/docx file to show us that you actually ran them yourself. This file will contain a list of different inputs and their results in one big file (similar to the sample runs shown in this document). You

must copy the console outputs one run after another in the **same order as your list of test cases** in the txt/doc/docx file.

## What to hand in

When you are done with this project, submit all your work through CatCourses.

***Before*** you submit, make sure you have done the following:

- Completed `Shop.java`.
- Listed your test cases for each requirement in `Test_cases.txt/doc/docx`.
- Placed your console output from all your runs in the same order as your list of test cases in `Test_cases.log`.
- Attached the `Shop.java`, `Test_cases.txt/doc/docx`, and `Test_cases.log` files.
- Filled in your collaborator's name in the "Comments…" text-box (if any) at the submission page.