Project 01: Shop Program
CSE-021-01 **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 during the lab sessions. Your partner can be in any section. Both you and your partner will make separate submissions and demonstrate individually, each noting the collaborator/partner in the "Comments…" textbox in the submission page.

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. Set up shop
   2. Buy
   3. List Items
   4. Checkout
Please type the function you want:
```

I. Setup Shop

1) Ask for the number of items to sell.
2) For each item:
   a) Ask for the name of the item (one word).
   b) Ask for the price (per package) of the item.
   c) Number of packages `x` to qualify for Special Discount: Buy `x` packages, get 1 free. This is a positive whole number as items are sold as multiples of packages.
3) Additional Discount
   a) Ask for the threshold – dollar value of purchase over which to offer additional discount.
   b) Ask for the rate (how much % discount). This must be a number greater than 0 but less or equal to than 0.5. If not, display an error message and ask for the input again [cf. Sample Run 2, Lines 20/21].
4) User can run setup multiple times at any point in the program, 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 (number of packages) they wish to purchase. This must be a number greater than or equal to 0. If not display an error message and ask the input again.
3) Customer can purchase multiple times, so only process the latest order.

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, price per package, and price*amount.

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) Original Sub Total: price* amount of each item purchased added together.
   b) Special Discount Total: total price of free packages based on special discounts added together.
   c) New Sub Total: (Original Sub Total) – (Special Discounts Total).
   d) Additional Discount: percentage of New Sub Total based on rate input during shop setup.
   e) Final Total: (New Sub Total) – (Additional Discount).
4) Ask the user if they would like to repeat the program, and if not end the program. Otherwise, repeat the program from the beginning.

Discount Calculation

Special Discount: As mentioned earlier, the program will ask for number of packages `x` to qualify for Special Discount. Suppose an item `alpha` is being sold at the price of `$10/pkg`. Suppose the user enters `x = 3` indicating that buy buying `3` packages gets `1` package for free. Note that you will need to maintain such numbers for each item being sold, so you may want to store them in an array (use a more meaningful name for this array than `x`). Now if the user buys `12` packages of `alpha`, then `3` will be free as can be seen below.

Pkg number	1	2	3	4	5	6	7	8	9	10	11	12
Free				✓				✓				✓

Multiplying the number of free packages with the price per package will give us the special discount on the item. In this case, the special discount for `alpha` will be `$30`. Your program will need to calculate the special discount for all items that have the number of packages to qualify for Special Discount set to greater than 0, and finally add them all up to get the Special Discount Total.

Additional Discount: After calculating the New Sub Total as (Original Sub Total) – (Special Discounts Total), check whether the New Sub Total is greater than the additional discount threshold. If it is, compute the additional discount as the (Additional Discount Rate) * (New Sub Total), where the Additional Discount Rate is a number between 0 and 0.5 signifying the percentage (this rate is entered by the user in the 3rd step of Setup Shop described earlier).

Important Additional Instructions

Your program will ask for the names of the items using the following prompts:

```
Enter the name of the 1st product:
…
Enter the name of the 2nd product:
…
Enter the name of the 3rd product:
…
```

To display the numbers as `1st`, `2nd`, `3rd`, etc. you will need to invoke the following function in your print statement:

```
public static String numSuffix(int i) {
	int rem = i % 10;
	switch (rem) {
		case 0:
		case 4:
		case 5:
		case 6:
		case 7:
		case 8:
		case 9:
			return (i + "th");
		case 1:
			if (i % 100 != 11)
				return (i + "st");
			else
				return (i + "th");
		case 2:
			if (i % 100 != 12)
				return (i + "nd");
			else
				return (i + "th");
		case 3:
			if (i % 100 != 13)
				return (i + "rd");
			else
				return (i + "th");
		default:
			break;
	}
	return "";
}
```

As you can see, this method takes an `int` parameter and returns a `String`. Calling `numSuffix(5)` will return the String `5th` and so on.

Specification Compliance: The following are some additional instructions to make sure your project complies with specifications:

1. Your program must produce an output that **exactly resembles the Sample Output shown below, including identical wording of prompts, spacing, input locations, etc**.
2. Your program must ensure the following:
   a. Your program should not display items that are not bought (amount is 0) when listing them out [cf. Sample Run 2, Lines 50-51]. If no items are purchased, then a special message is displayed [cf. Sample Run 3, Line 39]. Also, when listing items bought, items with singular packages must say `package` (and not `packages`) in the corresponding output statements [cf. Sample Run 3, Lines 59/60/61].
   b. The prompt to enter the additional discount rate is only shown when the threshold is greater than 0 [cf. Sample Run 3, Line 19].
   c. If no special or additional discounts are applied, a special message is displayed [cf. Sample Run 3, Lines 71/73].
3. Your program should correctly operate for all valid sequences of option choices. For instance, the user may setup the shop, buy items, list them out, and go back to setting up the shop again, resetting everything. In this case, you must make sure that choosing option 3 or 4 doesn't produce output corresponding to the previously bought items. In another valid run, the user may setup shop, buy items, list them out, and then go back to change the items bought. In this case only the last set of items bought are retained. **Make sure you test your program for all possible valid sequences of option choices**.

4. Before you submit, in Eclipse, type CTRL-A (to select everything) followed by a CTRL-I (to fix indentation) on BobcatHotel.java. In MacOS the corresponding keystrokes are Cmd-A followed by Cmd-I.

Sample Behavior (user input shown in green)

Sample Run 1: 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 (the full run is not shown):

```
This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 3

The shop has not been set up yet!

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

The shop has not been set up yet!

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

The shop has not been set up yet!

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please choose the function you want: 1
Please enter the number of items to set up shop: ...
```

Sample Run 2: 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. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 1
Please enter the number of items to set up shop: 3

Enter the name of the 1st product: alpha
Enter the per package price of alpha: 10
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for alpha, or 0 if no Special Discount offered: 3
```

```
Enter the name of the 2nd product: beta
Enter the per package price of beta: 20
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for beta, or 0 if no Special Discount offered: 0
Enter the name of the 3rd product: gamma
Enter the per package price of gamma: 30
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for gamma, or 0 if no Special Discount offered: 4

Enter the dollar amount to qualify for Additional Discount (or 0 if none offered): 100
Enter the Additional Discount rate (e.g., 0.1 for 10%): 3
Invalid input. Enter a value > 0 and <= 0.5: .2

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 3

You have not bought anything!

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 2

Enter the number of alpha packages to buy: 10
Enter the number of beta packages to buy: 0
Enter the number of gamma packages to buy: 20

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 3

10 packages of alpha @ $10.00 per pkg = $100.00
20 packages of gamma @ $30.00 per pkg = $600.00

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 4

Original Sub Total:       $700.00
Special Discounts:       -$140.00
New Sub Total:            $560.00
Additional 20% Discount: -$112.00
Final Sub Total:          $448.00

Thanks for coming!

-------------------------------------------------
Would you like to re-run (1 for yes, 0 for no)? 0
-------------------------------------------------
```

Explanation of the checkout values: In the sample run above, `alpha` is offered with buy 3 get 1 free special discount and `gamma` is offered with buy 4 get 1 free special discount. So, with 10 packages of `alpha`, the user gets 2 packages for free, and with 20 packages of `gamma` the user gets 4 packages for free. This gives us a total special discount of (2*10) + (4*30) = 140. Subtracting this special discount of 140 from the Original Sub Total of 700 gives us 560 as our New Sub Total. Now, since 560 is greater than our threshold of 100, we apply the additional discount rate of 20% on 560 to get 112. Finally subtracting 112 from the New Sub Total of 560 gives us 448 as our Final Sub Total.

Sample Run 3: Finally, the following is a (rather long) sample run when a customer purchases multiple times, runs the shop multiple times, goes back and resets the shop, and changes the number of items bought:

```
This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 1
Please enter the number of items to set up shop: 3

Enter the name of the 1st product: alpha
Enter the per package price of alpha: 10
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for alpha, or 0 if no Special Discount offered: 1
Enter the name of the 2nd product: beta
Enter the per package price of beta: 20
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for beta, or 0 if no Special Discount offered: 2
Enter the name of the 3rd product: gamma
Enter the per package price of gamma: 30
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for gamma, or 0 if no Special Discount offered: 3

Enter the dollar amount to qualify for Additional Discount (or 0 if none offered): 0

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 2

Enter the number of alpha packages to buy: 0
Enter the number of beta packages to buy: 0
Enter the number of gamma packages to buy: 0

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 3

No items were purchased.

This program supports 4 functions:
     1. Set up shop
     2. Buy
```

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

Enter the number of alpha packages to buy: 1
Enter the number of beta packages to buy: 1
Enter the number of gamma packages to buy: 1

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 3

1 package of alpha @ $10.00 per pkg = $10.00
1 package of beta @ $20.00 per pkg = $20.00
1 package of gamma @ $30.00 per pkg = $30.00

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 4

Original Sub Total:              $60.00
No Special Discounts applied
New Sub Total:                   $60.00
You did not qualify for an Additional Discount
Final Sub Total:                 $60.00

Thanks for coming!

-------------------------------------------------
Would you like to re-run (1 for yes, 0 for no)? 1
-------------------------------------------------

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 2

The shop has not been set up yet!

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 1
Please enter the number of items to set up shop: 2

Enter the name of the 1st product: gamma
Enter the per package price of gamma: 40
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for gamma, or 0 if no Special Discount offered: 4
Enter the name of the 2nd product: delta
Enter the per package price of delta: 50
```

```
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for delta, or 0 if no Special Discount offered: 5

Enter the dollar amount to qualify for Additional Discount (or 0 if none offered): 200
Enter the Additional Discount rate (e.g., 0.1 for 10%): .3

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 2

Enter the number of gamma packages to buy: -4
Invalid input. Enter a value >= 0: -2
Invalid input. Enter a value >= 0: 12
Enter the number of delta packages to buy: -5
Invalid input. Enter a value >= 0: 15

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 1
Please enter the number of items to set up shop: 3

Enter the name of the 1st product: gamma
Enter the per package price of gamma: 11.50
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for gamma, or 0 if no Special Discount offered: 4
Enter the name of the 2nd product: delta
Enter the per package price of delta: 14.43
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for delta, or 0 if no Special Discount offered: 5
Enter the name of the 3rd product: epsilon
Enter the per package price of epsilon: 2.25
Enter the number of packages ('x') to qualify for Special Discount (buy 'x' get 1 free)
for epsilon, or 0 if no Special Discount offered: 6

Enter the dollar amount to qualify for Additional Discount (or 0 if none offered): 100
Enter the Additional Discount rate (e.g., 0.1 for 10%): .25

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 2

Enter the number of gamma packages to buy: 12
Enter the number of delta packages to buy: 14
Enter the number of epsilon packages to buy: 20

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 3

```

```
12 packages of gamma @ $11.50 per pkg = $138.00
14 packages of delta @ $14.43 per pkg = $202.02
20 packages of epsilon @ $2.25 per pkg = $45.00

This program supports 4 functions:
     1. Set up shop
     2. Buy
     3. List Items
     4. Checkout
Please type the function you want: 4

Original Sub Total:       $385.02
Special Discounts:       -$56.36
New Sub Total:            $328.66
Additional 25% Discount: -$82.16
Final Sub Total:          $246.49

Thanks for coming!

------------------------------------------------
Would you like to re-run (1 for yes, 0 for no)? 0
------------------------------------------------
```

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`.
- Attached `Test_cases.txt/doc/docx` and `Test_cases.log` files containing the test cases and corresponding console outputs.
- Attached the `Shop.java` file.
- Filled in your collaborator's name (if any) in the "Comments…" text-box (if any) at the submission page.

Also, remember to demonstrate your code to the TA or instructor before the end of the demo period.