

**CSCI 1301 – Programming Principles I**  
**Georgia Southern University**  
**Department of Computer Science**  
**Fall 2024**

**Assignment 3**

**Point Value: 20 points**

**Due: Friday September 13, 2024, start of lab**

**NOTE:** Gradescope will give input test cases for any problems that require it. Use the Scanner's `nextDouble()`, `nextInt()`, etc. methods discussed in class to prompt the user for input and Gradescope will act as the user when your program requests it. In other words, just code user input like normal—as if a human were going to test your program.

**Description**

Write a Java program that modifies and improves Programming Assignment 1a. This version should disallow zero quantity orders (management has decided that everyone must buy at least one sandwich and at least one beverage for supply cost reasons).

Assume the convenience store only sells two items: beverages at \$0.50 each and sandwiches at \$2.75 each. Prices can be hard-coded, but should be set/stored in variables, and those variables should be used in all calculations.

Prompt the user for the number of beverages and store the entered quantity as a decimal number. If the quantity is positive, display the quantity and which item was ordered. If a positive quantity is entered, output product and quantity. If the user enters a zero, display an error message and ask the user, **only once**, for another quantity for that item. Prompt the user for the number of sandwiches and store the entered quantity. If a positive quantity is entered, output the product and quantity. If the user enters a zero, display an error message and ask the user, **only once**, for another quantity for that item.

Only display the order total if the number of sandwiches is greater than zero and the number of beverages is greater than zero. This should be executed with a **nested-if** statement or using logical operators.

If both of those conditions are not met, display a message saying that management requires at least one of each item being ordered. If they are met, display the subtotal, then the total with a 7.5% sales tax added (do not worry about any extra decimals for now, *as ugly as it will be*).

**HINTS/NOTES**

Drawing a flowchart for the required behavior of this program will help you identify what types of conditional statements to use, and when to use those statements. Pay attention to the sample runs provided in the expected output and how they operate.

Depending on the tax calculation used, a differing number of decimal places may appear in output. The method used for the expected output calculates percentage increase/decrease as a multiplication times 1.X with X being the percentage (so something like  $\text{tot} = \text{sub} * 1.15$  for 15% as opposed to  $\text{tot} = \text{sub} + (\text{sub} * .15)$ ).

If you are not using that style, it is likely going to involve differing decimal places. The other test cases will not execute if any fail, so you should use the above tax calculation method.

**Do not try to oversimplify output in this problem. There will be times you may need to repeat code to make this assignment work as shown.**

There are multiple test cases that your code should pass if it handles everything correctly. There are cases which produce a valid order, and cases that should produce error(s). For simplicity, the inputs are grouped together in the Expected Output sections below:

**Expected Output, valid input (do NOT hard code your output):**

Input: 5 3 (shown), or 0 5 3, or 5 0 3

Enter the number of beverages: 5  
Ordered: 5.0 beverages

Enter the number of sandwiches: 3  
Ordered: 3.0 sandwiches

The subtotal of 5.0 beverages and 3.0 sandwiches is \$10.75.  
With tax, the total is \$11.55625.

===== DIFFERENT TEST RUN BELOW =====

Input: 5 3, or 0 5 3 (shown), or 5 0 3

Enter the number of beverages: 0  
ERROR: A quantity of zero is not allowed.

Enter the number of beverages: 5  
Ordered: 5.0 beverages

Enter the number of sandwiches: 3  
Ordered: 3.0 sandwiches

The subtotal of 5.0 beverages and 3.0 sandwiches is \$10.75.  
With tax, the total is \$11.55625.

===== DIFFERENT TEST RUN BELOW =====

**Expected Output, invalid input (do NOT hard code your output):**

Input: 0 0 3 (shown), or 5 0 0, or 0 0 0 0

Enter the number of beverages: 0

ERROR: A quantity of zero is not allowed.

Enter the number of beverages: 0

Ordered: 0.0 beverages

Enter the number of sandwiches: 3

Ordered: 3.0 sandwiches

Your order total could not be calculated due to a zero quantity for an item.

**Code**

Use the provided template for this assignment. Make any necessary modifications to classes and class headers to complete this assignment.

```
public class PAssign03 {  
    public static void main(String[] args) {  
        // add your code here  
    }  
}
```

**Deliverables**

Name your program PAssign03.java. Programming Assignment 3 is to be individual work. Submit the program by the specified due date. Submit each file to its corresponding assignment on Gradescope.

See and follow the Programming Assignment Format document for submission requirements.

Use a utility similar to <https://www.diffchecker.com/> and the Expected Output to compare your program's output with the requested output as well as the unit tests provided within Gradescope. Programming is in the details, so double check punctuation, spacing, and case if your output does not match. When copying and pasting, be aware that Microsoft Word sometimes replaces normal quotes with Smart Quotes, which may need to be edited.

Last modified: September 3, 2024