

Purpose:

The purpose of this assignment is to practice user input, string variables, selection statements and algorithm development.

Teams:

The assignment can be done individually or in teams of two from the **same** section. Submit one assignment per team.

General Guidelines When Writing Programs:

- Include the following comments at the top of your source codes
// -----
// Assignment (include number)
// Question: (include question/part number, if applicable)
// Written by: (include your name(s) and student ID(s))
// -----
- In a comment, give a general explanation of what your program does. As the programming questions get more complex, the explanations will get lengthier.
- Include comments in your program describing the main steps in your program.
- Display a welcome message which includes your name(s).
- Display clear prompts for users when you are expecting the user to enter data from the keyboard.
- All output should be displayed with clear messages and in an easy to read format.
- End your program with a closing message so that the user knows that the program has terminated.

In this assignment you are required to write a Cashier Calculator, which will compute the total price of some items including taxes (PST & QST).

The sub-total, taxes, and total price are calculated as follows:

SubTotal = $\sum_{i=1}^n P(i)$ where n = number of items bought and P(i) = the price of item i.

GST = **Subtotal** × **Grate** where GRate is the GST tax rate (e.g., GRate=0.06 stands for 6%)

QST = (**SubTotal** + **GST**) × **QRate** where QRate is the QST tax rate (e.g., QRate=0.075 stands for 7.5%).

Note, unlike other provinces in Canada, QST is applied on top of the sum of the subtotal and GST.

Total = **SubTotal** + **GST** + **QST**

Your program will require the user to enter the following:

- 1) The number of items
- 2) The price of each item
- 3) The GRate in percentage
- 4) The QRate in percentage

The following constraints are given:

- The minimum number of items is 1 and the maximum number of items is 10.
- The minimum price of each item is \$1 and the maximum price of each item is \$1,000. The price may be a decimal number.
- The minimum tax rate for GST is 0 and the maximum tax rate for GST is 18. Suppose the tax rate is 12.345%, the user is expected to enter "12.345". You will have to divide the entered value by 100 to compute GRate. We assume that the user will not enter the "%" symbol.
- The minimum tax rate for QST is 0 and the maximum tax rate for QST is 22. Suppose the tax rate is 12.345%, the user is expected to enter "12.345". You will have to divide it by 100 to compute QRate. We assume that the user will not enter the "%" symbol.
- Note, all these minimum and maximum values are constants, i.e., they remain unchanged throughout the entire program. You should know how to define them.

The program must behave in the following manner:

- The program will require the user to enter the needed values one after another. If any of the entered values is invalid, then the program will detect that and go into a loop requiring the user to enter a valid value. The program must keep track of the total number of invalid values that the user has entered.
- Using SubTotal and GRate, the program should compute GST.
- Using GST and QRate, the program should compute QST.
- The program will calculate the SubTotal, GST, QST and Total then display the following:
 - How many times the user has entered invalid values for *all* the required inputs. If the user did not enter any invalid value, then no message should be displayed. In other words, "You have entered 0 invalid inputs" should NOT be displayed if the user has entered all values correctly.
 - The subtotal
 - The GST
 - The QST
 - The total

All output values must be formatted.

- The displayed subtotal, GST, QST, and total should have 2 decimal places. They have to align by their decimal point. The output should look like:

```

Please enter the number of items bought [1...10]: 11
Please enter the number of items bought [1...10]: 2
Please enter the price of item 1 [1...1000]: -5
Please enter the price of item 1 [1...1000]: 200
Please enter the price of item 2 [1...1000]: 600
Please enter the tax rate of GST in % [0...100]: 105
Please enter the tax rate of GST in % [0...100]: 6
Please enter the tax rate of QST in % [0...100]: -5
Please enter the tax rate of QST in % [0...100]: 7.5
You have entered 4 invalid inputs.
Sub Total:      $   800.00
GST:            $    48.00
QST:            $    63.60
Total:         $   911.60

```

Submitting Assignment 2

- **IMPORTANT:** You are allowed to work on a group of two (i.e. a group can have a maximum of two students). If your work on a team, **ONLY** one copy of the assignment is to be submitted for both members.
- Zip together the source codes. (Please use WINZIP).
- Naming convention for zip file: Create one zip file, containing all source files for your assignment using the following naming convention:
 The zip file should be called *a#_studentID*, where # is the number of the assignment *studentID* is your student ID(s) number. For example, for the first assignment, student 1234567 would submit a zip file named *a1_1234567.zip*. If you work on a team and your IDs are 1234567 and 4567890, you would submit a zip file named *a1_1234567_4567890.zip*.
- Submit your zip file at: <https://fis.encs.concordia.ca/eas/> as **Programming Assignment** and submission **#2**. Assignments submitted to the wrong directory would be discarded and no replacement submission will be allowed.
- Submit only ONE version of an assignment. If more than one version is submitted the first one will be graded and all others will be disregarded.

Evaluation Criteria for Assignment 2 (10 points)

| | |
|---|----------|
| Comments/description of variables/description of the steps in code/purpose of program | 1 point |
| Choice of variable names | 1 point |
| Indentation and readability of program | 1 point |
| Format/clarity/completeness/accuracy of output | 1 point |
| Proper use of required Java concepts/correctness of program | 6 points |