2022 Spring -- CSCI 1300L
Lab 05: Programming with loops

Introduction

In this lab, you will create a Python program called "Total Cost Calculator". To complete this assignment, you should properly use a loop structure in Python. Although you have been introduced with different types of loops and more than one way to exit the loop, it is recommended that you make the code efficiently, more structured, and with the best programming practice.

Lab objectives

After completing this lab, you will be able to:
(1) use Python to do calculations further,
(2) use a loop structure to implement the Python code,
(3) understand the differences between multiple types of loops, and
(4) know how to write the proper code to exit the loop for a program.

Assignment

Assume that you are shopping or managing the inventory, the program will ask you to input each item's unit price and each item's quantity. After you have added the information of all items, the program will calculate the total cost for you.

The math of this total cost calculator is as follows:

*Total cost = P1 * Q1 + P2 * Q2 + P3 * Q3 + ... + Pn * Qn*

where P1, P2, P3, ... Pn are the unit price for each item, Q1, Q2, Q3, ... Qn are the quantity for each item.

Since the program does not know how many items you have or will add, you will be asked by the program to indicate if you have finished adding items. Think how to store "intermediate values" and add them into the total. Do you have to use 100+ variables with different names?

The program should perform its functions in the following way:

Program asks: Item unit price?
User inputs: aa.aa
Program asks: Quantity = ?
User inputs: bb
Program asks: finished adding?
User inputs: y or n

If the user finishes adding items, the program will exit the loop. At the end of the program, there should be an output of the total price for all items. If the user does not finish adding items, the program will stay in the loop by asking the user to enter another item.

Take an example of the program execution. Highlighted texts are user inputs.

```
Total Cost Calculator
What is the item's unit price?
3.50
What is the quantity of that item?
10
Any more items to add?
y
What is the item's unit price?
4
What is the quantity of that item?
9
Any more items to add?
n
The total cost is 71.00
```

Since the focus of this assignment is the use of loops in the program, you do not need to worry about any user inputs that do not make sense, such as a negative value or a non-integer of quantity. As long as the item's unit price is treated to accept a number with digits after the decimal point, it will be good at this time. Due to the imprecision issue of floating-point numbers, the display of the total cost does not have to be "exactly" precise. As long as the difference between the calculated/displayed value and the actual value is within 0.01, the calculation is deemed correct.

Your program's output does not need to be exactly the same as the example output. However, your program must:
(1) contain a loop structure,
(2) handle all three user inputs in the loop,
(3) be able to end by exiting the loop under certain conditions (i.e. no infinite loop allowed),
(4) display the relevant message so that the user is notified what to see/do next, and
(5) correctly calculate and display the total cost based on all previous inputs.

Submission instruction

After you have completed the assignment, upload and submit the Python source code file *Lab05.py* to eLC. Always double check that your submission was successful on eLC.

Grading

A score between 0 and 5 will be assigned.

1. The program can successfully accept an end user's inputs. (1 point)
2. The program can successfully process "one item" (i.e. the initial round of the loop) and its calculation of total cost is correct. (1 point)
3. The program can successfully process "two or more items" (i.e. subsequent rounds of the loop) and its calculation of the total cost is correct. (1 point)
4. The program can finish with a proper loop exit condition. (1 point)
5. Only a single source code file is submitted and no other file is submitted (0.5) and the entire Python program can be executed without any additional error (0.5). (1 point)

Special notice regarding the submission:

Late submission penalty. Points will be deducted from the original grade. If your submission is after the posted deadline…

(1) within 24 hours: -2
(2) between 24 hours and 48 hours: -3
(3) between 48 hours and 72 hours: -4
(4) after 72 hours: assignment will not be accepted.