CSE 231 Summer 2022

Programming Project 02

This assignment is worth 65 points (6.5% of the course grade) and must be **completed and turned in before 11:59 on Tuesday, May 31, 2022.**

Assignment Overview

This assignment will give you more experience on the use of:

1. integers (`int`)
2. floats (`float`)
3. conditionals
4. iteration

Your program will calculate change. It will start with a stock of coins. It will then repeatedly request the price for an item to be purchased or to quit. If a price is input, it will prompt for dollars in payment and print the change due using the minimum number of coins possible. Before quitting, it will display the value of coins remaining in the stock. An example interaction with our program appears at the end of this description.

Background

The algorithm for calculating the numbers of coins of each denomination to dispense so as to minimize the total number of coins is an example of a *greedy* algorithm. You start by figuring out the most number of quarters you can dispense (without exceeding the amount), then the most number of dimes, then the number of nickels and finally pennies. If you are curious, you can read about the Greedy Algorithm.

Project Description / Specification

Your program must meet the following specifications:

1. At program start, assume a stock of 10 nickels, 10 dimes, 10 quarters, and 10 pennies.
2. Repeatedly prompt the user for a price in the form `xx.xx`, where `x` denotes a digit, or to enter '`q`' to quit.
3. When a price is entered:
   a. If the price entered is negative, print an error message and start over requesting either a new price or to quit (indicated by entering a 'q').
   b. Prompt for the number of dollars in payment. If the payment is insufficient, print an error message and reprompt for payment.
   c. Next determine the coins to be dispensed as change. This calculation will depend on the amount to be dispensed and also on the number of coins left in the stock. For example, the least number of coins needed to make change of $1.30 is 6: 5 quarters and 1 nickel. But if there are only 3 quarters, 3 dimes, and 10 nickels left in the stock, then the least number is 11: 3 quarters, 3 dimes, and 5 nickels.
   d. Print the numbers of the coins to be dispensed as change and their denominations. (Omit a denomination if no coins of that denomination will be dispensed.)
   e. In case exact payment is made, print a message such as `"No change."`
   f. If the change cannot be made up with the coins remaining, print an error message and halt the program
4. Just before quitting, print the total amount (the number of dollars and number of cents) left in the stock.

Deliverables

The deliverable for this assignment is the following file:

`proj02.py` -- your source code solution

Be sure to use the specified file name and to submit it for grading via Coding Rooms before the project deadline

Notes and Hints:

1. To clarify the project specifications, sample output is appended to the end of this document.
2. Items 1-6 of the Coding Standard will be enforced for this project.
3. We provide a `proj02.py` program for you to start with. It has a simple `while` loop (notice how input is prompted before the loop and at the bottom of the loop.
4. Floating point numbers can be difficult to work with due to the imprecision of representing real numbers in a finite number of computer-memory bits. To avoid imprecision do your calculations in cents, i.e. as type `int`. For example, $1.15 is the same as 115 cents. To see the problem, try evaluating `1.15*100` in the Python shell and compare that to evaluating `round(1.15*100).`
5. There are many ways to calculate the maximum number of quarters to make change using the greedy algorithm.
   a. One is with a `while` loop where you keep subtracting 25 from the amount of change due and increment the number of quarters. End the loop when there are less than 25 cents due or you are out of quarters. After determining quarters you can determine dimes in the same way, e.g. using 10 instead of 25. Work your way down through the other coins.
   b. Alternatively, use the quotient (//) operation for integers for finding the numbers of each coin. For example, 195//25 is 7, the most number of quarters in 195 cents. However, be careful: if the stock has fewer than 7 quarters left, you will only be able to dispense the number left in the stock. For example, if there are only 6 quarters left, then you can dispense only 6 quarters and must use dimes and nickels to make up any remaining change.
6. When we learn more about formatting strings, we can more easily and elegantly print monetary amounts. For now, just use the Python `print`(…) command with appropriate string and/or integer arguments to print the number of dollars and the number of cents.
7. One tricky control issue is how to end the program if you run out of coins in the stock.
   Hint: an `empty_stock` Boolean variable. The problem is that if you use break, `break` only breaks out of the current, innermost loop whereas you may be within multiple loops so when you `break` you can set the `empty_stock` Boolean to `True` and use that to break out of the enclosing loops.
8. You do not need to check for any input errors other than those mentioned in this description.
9. You may not use advanced data structures such as lists, dictionaries, sets or classes in solving this problem.

Sample Interaction:

Test 1

```
Welcome to change-making program.

Stock: 10 quarters, 10 dimes, 10 nickels, and 10 pennies
Enter the purchase price (xx.xx) or 'q' to quit: 1.5

Input dollars paid (int): 2

Collect change below:
Quarters: 2

Stock: 8 quarters, 10 dimes, 10 nickels, and 10 pennies

Enter the purchase price (xx.xx) or 'q' to quit: q
```

Test 2

```
Welcome to change-making program.

Stock: 10 quarters, 10 dimes, 10 nickels, and 10 pennies
Enter the purchase price (xx.xx) or 'q' to quit: 2

Input dollars paid (int): 2
No change.

Stock: 10 quarters, 10 dimes, 10 nickels, and 10 pennies

Enter the purchase price (xx.xx) or 'q' to quit: q
```

Test 3

```
Welcome to change-making program.

Stock: 10 quarters, 10 dimes, 10 nickels, and 10 pennies
Enter the purchase price (xx.xx) or 'q' to quit: 1.5

Input dollars paid (int): 5

Collect change below:
Quarters: 10
Dimes: 10

Stock: 0 quarters, 0 dimes, 10 nickels, and 10 pennies
Enter the purchase price (xx.xx) or 'q' to quit: 0.5

Input dollars paid (int): 2

Error: ran out of coins.
```

Test 4

```
Welcome to the change-making program.

Enter the purchase price (xx.xx) or `q' to quit: -1.20
Error: purchase price must be non-negative.

Enter the purchase price (xx.xx) or `q' to quit: 1.43

Input dollars paid (int): 1
Error: insufficient payment.

Input dollars paid (int): 2

Collect change below:
Quarters: 2
Nickels: 1
Pennies: 2

Stock: 8 quarters, 10 dimes, 9 nickels, and 8 pennies

Enter the purchase price (xx.xx) or `q' to quit: 0.13

Input dollars paid (int): 1

Collect change below:
Quarters: 3
Dimes: 1
Pennies: 2

Stock: 5 quarters, 9 dimes, 9 nickels, and 6 pennies

Enter the purchase price (xx.xx) or `q' to quit: 2.22

Input dollars paid (int): 4

Collect change below:
Quarters: 5
Dimes: 5
Pennies: 3

Stock: 0 quarters, 4 dimes, 9 nickels, and 3 pennies

Enter the purchase price (xx.xx) or `q' to quit: q
```

Test 5
Blind test

Scoring Rubric

```
Computer Project #02                                    Scoring Summary

General Requirements

______    5 pts    Coding Standard
        (descriptive comments, mnemonic identifiers, format, etc...)


Implementation:

__0__    (12 pts)   Test Case 1:
__0__    (12 pts)   Test Case 2:
__0__    (12 pts)   Test Case 3:
__0__    (12 pts)   Test Case 4:
__0__    (12 pts)   Test Case 5: Blind Test


TA Comments:
```