

## Project 3

### Vending Machine Change Maker

#### Assignment Objectives

This assignment will allow you to apply iteration in your code and will allow you to become more experienced in using integers, floats, strings, and conditional statements.

Your program will simulate a simple **change-maker for a vending machine**.

The vending machine will start with a stock of **coins and dollars**. Then it will **repeatedly request** the user purchase an item (enter the price for an item to be purchased) or to quit.

- If the user chooses to **enter a price**, the vending machine will accept payment in nickels, dimes, quarters, one-dollar and five-dollar bills deposited one at a time.
  - When the user has deposited enough to cover the cost of the item, the program will **calculate the coins to be dispensed in change**.
  - Alternately, the user **can cancel the purchase** up until full payment has been deposited. In that case, **your program will calculate the coins to be dispensed to refund any partial payment already deposited**.
  - With **each purchase**, the program will update the **stock of coins and dollars**.
- If the user chooses to **quit**. Our vending machine will output the remaining stock of coins and dollars.

More detailed instructions are provided with the interactive example below.

All change and refunds must be in **coins only** and must use the **minimum number of coins possible**.

#### Background

The algorithm for calculating the numbers of coins of each denomination to dispense that uses **minimum number of coins possible (coins only)** is an example of a *greedy* algorithm. Since each coin is at least twice the value of the previous coin, you can start by

- figuring out the greatest number of quarters you can dispense (without exceeding the number of quarters available),
- then the greatest number of dimes, and
- then the number of nickels.

Knowledge of greedy algorithms is not required to complete this assignment. However, if you would like to read more about greedy algorithms, please check the following link:

[http://en.wikipedia.org/wiki/Greedy\\_algorithm](http://en.wikipedia.org/wiki/Greedy_algorithm).

#### Project Description / Specification

Your program must meet the following specifications:

1. Your program should start with a stock of **25 nickels, 25 dimes, and 25 quarters**.
  - a. Print the contents of the stock.
2. Repeatedly prompt the user for a price of the item they want to purchase in form xx.xx, where x denotes a digit, or to enter 'q' to quit.
3. When a price is entered:

- a. Check that the price entered is a **(non-negative) multiple of .05** (i.e., it is payable in nickels). Otherwise (if not), then **print an error message and start over requesting either a new price or to quit** (indicated by entering a 'q').
  - b. Print a **menu for indicating the coin/bill deposited** or to **cancel the payment (by entering 'c')**.
  - c. Prompt for a **selection from this menu**.
  - d. If the user enters an **illegal selection, re-prompt for the correct one**.
  - e. Following **each deposit, print the remaining amount owed** (indicate the number of dollars and the number of cents in \$xx.xx format).
  - f. When full payment has been deposited or a 'c' has been entered, determine the **coins to be dispensed in change or a refund**. This calculation will depend on the amount to be dispensed and on the number of coins left in the stock.  
For example, the least number of coins needed to make up \$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.
  - g. **Print the numbers of the coins** to be dispensed and their denominations. (Omit a denomination if no coins of that denomination will be dispensed.)
  - h. In case an **exact payment is deposited**, print a message such as “No change.”
  - i. If the **change cannot be made up with the coins remaining**, dispense the **coins available without exceeding the change amount** and indicate the amount still due to the user, which will have to be collected from a store manager.  
For example, if the stock contains one nickel, no dimes, and a quarter and if the change amount is 15 cents, dispense just the nickel and indicate the user should collect 10 cents from a store manager.
  - j. **Print the contents of the stock** following the transaction.
4. Just before quitting, print the total amount (the number of dollars and number of cents) left in the stock.

### Deliverables

1. You are to upload 1 file (The Project report) that includes:
  - a. A copy of your complete documented Python code
  - b. And a screenshot of the output of different runs that has all cases.
2. Be ready to demonstrate your code in the lab. -you will be informed by your lab instructor-

### Notes and Hints:

1. Floating-point numbers can be difficult to work with due to imprecision. To avoid imprecision in this program, you can multiply the price by 100, round, and convert it to an integer (number of cents).  
For example, \$1.15 is the same as 115 cents.  
To see why you need to round, try evaluating `1.15*100` in the Python shell.  
Now evaluate `round(1.15*100)`.
2. The quotient (`//`) operation for integers will be useful for finding the numbers of each coin.
  - For example, `195//25` is 7, the most number of quarters in 195 cents.
    - But 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.
3. When we learn about format 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.
4. You do not need to check for any input errors other than those mentioned in this description.
5. The logic for this project is sufficiently complex that you will need to break it into small steps that you can implement incrementally.  
After implementing each step, test it thoroughly, solve it as if it is one full project (subproblem) before trying to implement the next step.  
For example, you could:

- First implement the logic to initialize the stock, print it, and repeatedly prompt for a price until a 'q' is entered; just print the value that was input. Test and back up your code.
- Next, add code to check that the input is a legal price and, if it is not, to prompt for a new value. Again, just print the value that was input. Test and back up your code.
- Next, add code to print the menu of selections. Test and back up your code.
- Next, add code to repeatedly prompt the user to deposit a coin until the full amount is collected or a 'c' is entered and to print the remaining amount owed after each deposit. Test and back up your code.
- Next, add code to update the stock. Test and back up your code.
- Next, add code to calculate the total amount of change or refund to be dispensed.
- Next, add code to calculate the number of coins of each denomination to be dispensed and to update the stock.
- Etc.

It is not required that you develop the project exactly in this way. But if you don't find a way to incrementally build and test your projects, they generally will be more difficult and take you longer to do. I always recommend an **incremental strategy**, not just because it is easier, but also because it allows the grader to give partial credit. If you hand in a program that can run and that meets some, but not all, of the requirements, we can grade what you managed to get working.

#### Sample Interaction (user inputs shown in red):

```
>>> ===== RESTART =====
>>>
Welcome to the vending machine change maker program
Change maker initialized.
Stock contains:
    25 nickels
    25 dimes
    25 quarters
    0 ones
    0 fives

Enter the purchase price (xx.xx) or `q' to quit: 1.96
Illegal price: Must be a non-negative multiple of 5 cents.

Enter the purchase price (xx.xx) or `q' to quit: 1.95

Menu for deposits:
'n' - deposit a nickel
'd' - deposit a dime
'q' - deposit a quarter
'o' - deposit a one dollar bill
'f' - deposit a five dollar bill
'c' - cancel the purchase

Payment due: 1 dollar(s) and 95 cents
Indicate your deposit: 1
Illegal selection: 1
Payment due: 1 dollar(s) and 95 cents
Indicate your deposit: o
Payment due: 95 cents
```

Indicate your deposit: o

Please take the change below.

1 nickels

Stock contains:

24 nickels

25 dimes

25 quarters

2 ones

0 fives

Enter the purchase price (xx.xx) or `q' to quit: 3.25

Menu for deposits:

'n' - deposit a nickel

'd' - deposit a dime

'q' - deposit a quarter

'o' - deposit a one dollar bill

'f' - deposit a five dollar bill

'c' - cancel the purchase

Payment due: 3 dollars and 25 cents

Indicate your deposit: o

Payment due: 2 dollars and 25 cents

Indicate your deposit: d

Payment due: 2 dollars and 15 cents

Indicate your deposit: d

Payment due: 2 dollars and 5 cents

Indicate your deposit: o

Payment due: 1 dollars and 5 cents

Indicate your deposit: d

Payment due: 95 cents

Indicate your deposit: c

Please take the change below.

9 quarters

1 nickels

Stock contains:

23 nickels

28 dimes

16 quarters

4 ones

0 fives

Enter the purchase price (xx.xx) or `q' to quit: .05

Menu for deposits:

'n' - deposit a nickel

'd' - deposit a dime

'q' - deposit a quarter  
'o' - deposit a one dollar bill  
'f' - deposit a five dollar bill  
'c' - cancel the purchase

Payment due: 5 cents

Indicate your deposit: f

Please take the change below.

16 quarters

9 dimes

1 nickels

Stock contains:

22 nickels

19 dimes

0 quarters

4 ones

1 fives

Enter the purchase price (xx.xx) or 'q' to quit: 25

Menu for deposits:

'n' - deposit a nickel

'd' - deposit a dime

'q' - deposit a quarter

'o' - deposit a one dollar bill

'f' - deposit a five dollar bill

'c' - cancel the purchase

Payment due: 25 dollars and 0 cents

Indicate your deposit: f

Payment due: 20 dollars and 0 cents

Indicate your deposit: f

Payment due: 15 dollars and 0 cents

Indicate your deposit: f

Payment due: 10 dollars and 0 cents

Indicate your deposit: f

Payment due: 5 dollars and 0 cents

Indicate your deposit: c

Please take the change below.

19 dimes

22 nickels

Machine is out of change.

See store manager for remaining refund.

Amount due is: 17 dollars and 0 cents

Stock contains:

0 nickels

0 dimes

0 quarters  
4 ones  
5 fives

Enter the purchase price (xx.xx) or `q' to quit: .35

Menu for deposits:

'n' - deposit a nickel  
'd' - deposit a dime  
'q' - deposit a quarter  
'o' - deposit a one dollar bill  
'f' - deposit a five dollar bill  
'c' - cancel the purchase

Payment due: 35 cents

Indicate your deposit: q

Payment due: 10 cents

Indicate your deposit: d

Please take the change below.

No change due.

Stock contains:

0 nickels  
1 dimes  
1 quarters  
4 ones  
5 fives

Enter the purchase price (xx.xx) or `q' to quit: .35

Menu for deposits:

'n' - deposit a nickel  
'd' - deposit a dime  
'q' - deposit a quarter  
'o' - deposit a one dollar bill  
'f' - deposit a five dollar bill  
'c' - cancel the purchase

Payment due: 35 cents

Indicate your deposit: q

Payment due: 10 cents

Indicate your deposit: q

Please take the change below.

1 dimes

Machine is out of change.

See store manager for remaining refund.

Amount due is: 5 cents

Stock contains:

0 nickels  
0 dimes  
3 quarters  
4 ones  
5 fives

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

Total: 29 dollars and 75 cents  
>>>