What's For Sale?

Brayan Quevedo Ramos

Porterville College

Engineering: Introduction to Programming, 30214

Lab Report

07, February 2022

Contents

1 Introduction

This program revolves around user input and an electronic Point-Of-Sale (POS) system. What a POS system is, is an application where the user can add, modify, delete, and checkout within a confined space. This confined space typically revolves around one product category. In this instance, it's a McDonald's store. The system initially greets the user, to where the user can order items, and checkout. Of course, this is all fictional, but is supposed to represent a real system.

2 Requirements

Given by the assignment include the following:

1. Must introduce store and products to user.
2. Must have at least 3 different items to sell
3. Must ask for user input in regard to how many of each item the customer would like
4. Must add in tax to final price
5. Must relay back to the customer their order and price

3 Solution/Program Description

This program takes place inside the terminal of Python 3.8. There's a few workflows that coincide: Global Variables, Objects/Classes andHigh Level Navigation. With these functions in mind, what the user should see is an introduction, followed by a menu to order, ordering menu, and a checkout option!

3.1 Global Variables

Global Variables: There are two global variables that were utilized in the program, userOrder and userTotal.

```
#-------------Global variables-----------------
userOrder = [] # How many menuItems (str) have been allocated
userTotal = [] # Get itemPrice (float) for same index number of userOrder
```

What the purpose of these global variables are is to store order information that the user requests. So, if the user want's three items, they'd be appended to the string array, while the contents of the item would appear in the float array.
Since classes/objects are difficult to manipulate inside a multi-dimensional array in Python, there were two parallel arrays that were always linked together (by index). Whenever an append to the array would occur, there would be a call to both arrays so that data is universal for the checkout process. There is a difficulty curve to code readability using this as a result of linking arrays together and breaking industry norms.

3.2 Objects/Classes

Objects/Classes: To preserve runtime code execution, any menu items created must be created as a result of the MenuItem class.

```
#-------------Class variables-----------------
# CLASS MenuItem
# @CLASSargs: self (self), itemName (str), itemPrice (float)
# @CLASSspecial: Used to create all of the menu items for the Mcdonalds restarunt.
# @note:
class MenuItem:
    def __init__ (self, itemName, itemPrice, itemQuantity, itemNumber):
        self.itemName = itemName
        self.itemPrice = itemPrice
        self.itemQuantity = itemQuantity
        self.itemNumber = itemNumber

    def print(self):
        print(self.itemName, " ", self.itemPrice, " ", self.itemQuantity, " ", self.itemNumber)

    def getName(self):
        return self.itemName

    def getPrice(self):
        return self.itemPrice


cheeseburger = MenuItem("CheeseBurger", 1.99, 1, 100)
hamburger = MenuItem("Hamburger", 1.49, 1, 101)
chickenburger = MenuItem("McChicken", 2.49, 1, 102)
chickenNuggets = MenuItem("Chicken Nuggets (10 piece)", 4.99, 1, 103)
eggMcMuffin = MenuItem("EggMcMuffin", 3.50, 1, 104)
```

The design of this implementation is to quickly create as many objects (menuItem) as needed for the restaurant. This leads to a faster execution time, and more stability (less bugs) in scaling the program.
One of the overlying issues with having a global Class and Object is that since they aren't technically defined and stored in an array, it's impossible to create more items during runtime – what you define are the options they get. Furthermore, since Python doesn't have any included library for statistical search probabilities, ordering something via the menu is painful. Instead of allowing the user to type what they want in a natural language, they're prompted to: typing the index number of what item they want (100 + n) where n is item, followed by quantity of said items.

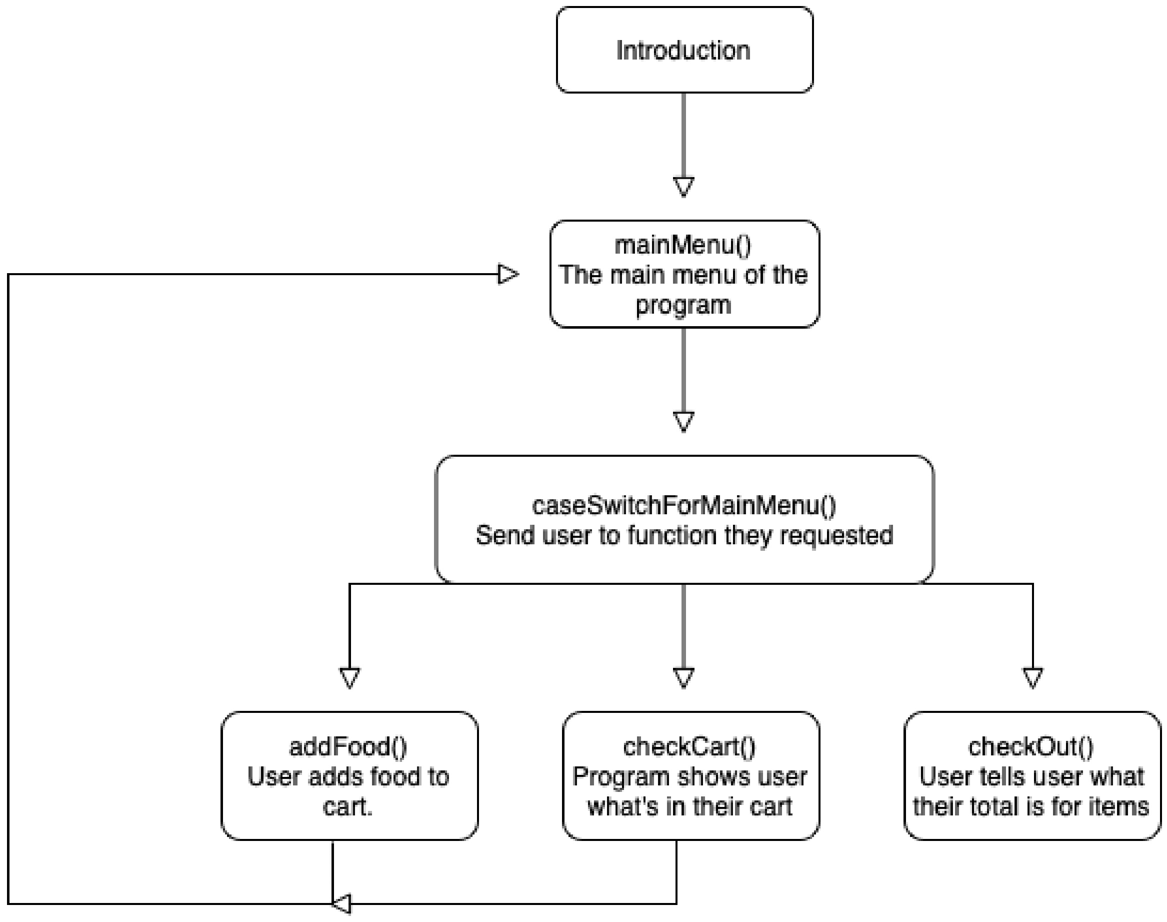
3.3 Navigation

High Level Navigation: One of the easiest parts to create and implement was the navigation. I say this because all the navigation did was take the user to one of three locations (listed in the Flowchart section), so it was difficult to create bugs. I did create an authentication function in the event the user didn't indicate a correct input, which prevents the program from bugging out. Once the user was redirected to the appropriate functions to complete the task (with the exception of checkOut) they'd be brought back to the navigation.

```
#-------------Function: mainMenu()------------
# @args: none
# @special: will be called by some function to continue loop.
# @timecomp: O(n)
#           - n: how many times user failed to select correct option
def mainMenu():
    print("How may I take your order?")
    print("[1]: Show Menu")
    print("[2]: Check Cart")
    print("[3]: Checkout")

    userMoveOn = False
    userChoice = 0

    while (userMoveOn == False):
        userChoice = eval(input("Please select an option: "))
        userMoveOn = checkUserAnswer(userChoice)

    caseSwitchForMainMenu(userChoice)


#-------------Function: checkUserAnswer()---------
# @args: userChoice (int)
# @special: Can only be called by mainMenu()
#           - Checks arg userChoice to see if correct input was made on the mainMenu() function
# @timecomp: O(3)
#           - 3: Worst case scnario where for the if statement does 3 checks to see if user
#                selected correct menu option!
def checkUserAnswer(userChoice):
    localUserChoice = userChoice
    if (localUserChoice == 1 or localUserChoice == 2 or localUserChoice == 3):
        # user has sucsessfully made a choice
        return True
    else:
        print("Sorry! I didn't understand that can you try again?")
        return False


#-------------Function: caseSwitchForMainMenu()-----
# @args: valueChosen (int)
# @special: Can only be called by mainMenu()
#           - Used to redirect user to one of three function [provided by mainMenu]
#           to the function that does what user wanted to do!
# @timecomp: O(3)
#           - 3: Worst case scnario where user wanted 3rd option from mainMenu()
#                therefore alg must make 3 comparisons.
def caseSwitchForMainMenu(valueChosen):
    if valueChosen == 1:
        showMenu()
    elif valueChosen == 2:
        checkCart()
    elif valueChosen == 3:
        checkOut()
    else:
        print("Whoops you're not supposed to be here")
```

4 Flowchart

Below is a high-level overview of how the program functions:


Introduction
mainMenu() The main menu of the program
caseSwitchForMainMenu() Send user to function they requested
addFood() User adds food to cart.
checkCart() Program shows user what's in their cart
checkOut() User tells user what their total is for items

5 Discussion

5.1 Known Bugs/Errors

From current testing, there are no ways to break the program.

5.2 Edge Cases

Due to some of the poorly designed algorithms, from testing of different edge cases, there seems to be no ways to break the program. The program expects very specific input, has conditions it requires, and will reiterate though function until it gets "said" input. There isn't a try catch exception in Python, but there is a while (arg) loop, so even though its hyper-inefficient, it's quick to code!

6 Lessons Learned

6.1 What Did I learn?

If I had to say that I learned something, it would have to be basic Python structure and syntax. You'll hear me say it once and you'll hear me say it again, but I am still not the most comfortable with Python – I know JavaScript and Java much better! I learned how to implement Objects/Classes, calling functions, and began to use return types as inputs for a function. I still don't know proper commenting structure nor syntax, (there isn't an ES6 like with JavaScript) where I know I'm writing programs in the most code-friendly way, but I am learning!
I would also like to learn hoe to NOT force every argument to be transitioned to a local variable before using it for whatever purpose. Default data-type error I got was tuple instead of int/float, so I must learn more on how to change that!
Finally, I'd like to learn better developmental practices, so whenever I make future programs I can make more optimized programs with better code readability. I felt like everything I wrote was very spaghetti and not clean, so if I could learn how to make anti spaghetti that would be amazing!

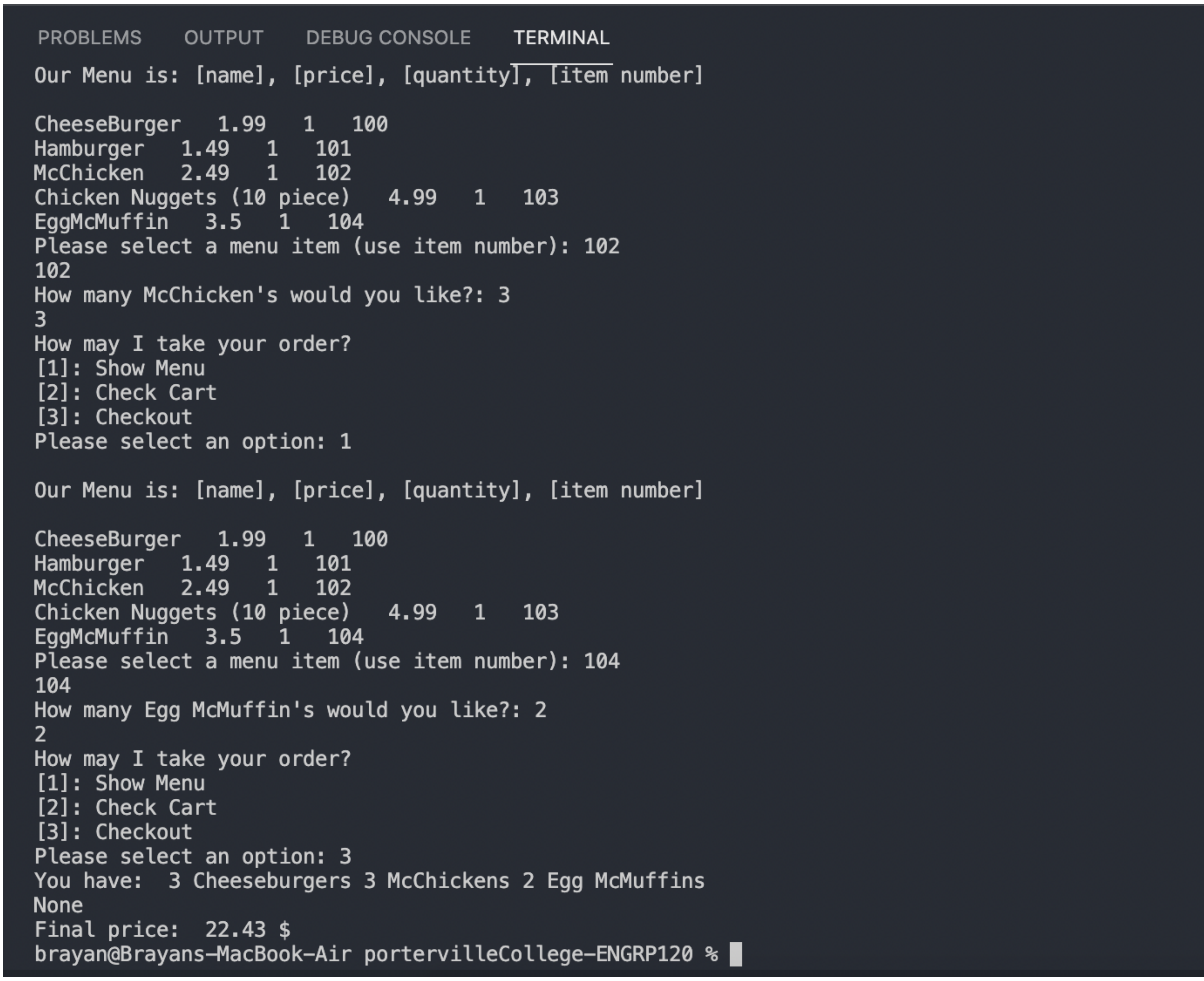
6.2 What I Would Do Differently?

To me, the architecture of my program was very garbage. If it's a static file, then it's terrible in my opinion. I didn't design a program where I can quickly add more objects and not change any other code. Perhaps it's because I'm rusty at design, or because I'm still fairly new to Python, but hopefully this changes in future projects!

6.3 Revision of Exercise

Something that I wasn't used to is to be given a test or data-set. In high-school, whenever I had to turn in an assignment, there would be a predetermined path that the user would take (that we had knowledge of) to test edge cases or not. For example, if the computer asked for two integers, there would be an input of a float and string to see how the program would handle bizarre responses. In bigger projects, there would be example inputs/out to see if our program was working correctly before we turned it in!

7 Results



```
Our Menu is: [name], [price], [quantity], [item number]

CheeseBurger   1.99   1   100
Hamburger   1.49   1   101
McChicken   2.49   1   102
Chicken Nuggets (10 piece)   4.99   1   103
EggMcMuffin   3.5   1   104
Please select a menu item (use item number): 102
102
How many McChicken's would you like?: 3
3
How may I take your order?
[1]: Show Menu
[2]: Check Cart
[3]: Checkout
Please select an option: 1

Our Menu is: [name], [price], [quantity], [item number]

CheeseBurger   1.99   1   100
Hamburger   1.49   1   101
McChicken   2.49   1   102
Chicken Nuggets (10 piece)   4.99   1   103
EggMcMuffin   3.5   1   104
Please select a menu item (use item number): 104
104
How many Egg McMuffin's would you like?: 2
2
How may I take your order?
[1]: Show Menu
[2]: Check Cart
[3]: Checkout
Please select an option: 3
You have:  3 Cheeseburgers 3 McChickens 2 Egg McMuffins
None
Final price:  22.43 $
brayan@Brayans-MacBook-Air portervilleCollege-ENGRP120 %
```