Restaurant Program - Final Project

Komal Khan IT 109

Problem Statement

- The application is trying to build a solution to help process food orders at Doordoor.com. It will present the user with a short menu with a list of choices and their corresponding costs.
- The user can add as many food selections as they want to their cart until they select option ‘c’ to Checkout. Then they are met with a receipt and askes if they would like to shop again with a new cart

Project Scope / Features

- The program offers 8 choices at the beginning, which can be chosen in any order.
  - Five food categories are offered, each with a subset of food dishes
- The other options include displaying what is currently in the cart and seeing the restaurants current special
- The final option is checking out, which displays the quantity and name of everything in the cart, as well as the price of each and the final price of the cart

```
1 - Appetizers
2 - Entrees
3 - Beverages
4 - Desserts
5 - Side Orders
d - Display Current Cart Contents
c - Checkout
s - Todays Special
Select one of the menu options above
5
a. Egg Roll Omelette $4
b. Gol Gappe $5
c. Elote Street Corn $6
```

implementation

- Conversion of ideas to executables began with beginning the rough code, running it, and then fixing bugs from there
- HW7 ⟹ HW8: Instead of the user inputting the quantity and cost, costs become set in the program as a tuple with its corresponding item. Submenus are added for each food category
- HW8 ⟹ HW9: new display function, session summary with string format()
- HW9 ⟹ HW10: from utilizing lists and dictionaries to utilizing files

```
def dispcart():
    print(f"\n{astr:<15}{str(aamt):>5}{'$'+str(acost):>10}",f"\n{estr:<15}{str(eamt):>5}{'$'+str(ecost):>10}",
          f"\n{bstr:<15}{str(bamt):>5}{'$'+str(bcost):>10}",f"\n{dstr:<15}{str(damt):>5}{'$'+str(dcost):>10}",
          f"\n{sstr:<15}{str(samt):>5}{'$'+str(scost):>10}",)
```

```
Select one of the menu options above
d

                0        $0
Japchae         1       $15
                0        $0
Churros         1        $6
                0        $0
```

```
txt = ' items bought\n {price:.2f} total '
fin = str(carts) + ' carts were processed' + '\n' + str(amount)
print(fin + txt.format(price = cost))
```

```
Would you like to restart and continue shopping? yes or nono
1 carts were processed
3 items bought
 28.00 total
```

```
def men(x):
    s = 'a. ' + x[0] + ' $' + str(x[1]) + '\n' + 'b. ' + x[2] + ' $' + str(x[3]) + '\n' + 'c. ' + x[4] + ' $' + str(x[5])
    return s
```

```
Select one of the menu options above
4
a. Songpyeon $8
b. Kheer $9
c. Churros $6
Which dessert would you like?: c
Add Churros to the cart (y/n)?: y
```

Menu items are retrieved from the file for each category

Formatting & Files - CODE

Hardware/Software Used

- `from numpy import random` – to randomly select a special for the restaurant and access it in the file
- `import logging, import datetime, open()` and `close()` – for the files to log the current applications date and time on file after the end of the program

```
import logging
import datetime
from numpy import random
```

```
log = open(r"C:\Users\komal\Downloads\IT109HA10Log.txt",'a')
x = datetime.datetime.now()
log.write(str(x))
log.write('\n' + 'Currently on cart number ' + str(carts) + '\n' +
          str(aamt+samt+eamt+bamt+damt) + ' items at a cost of $' + str(amount))
log.close()
```

Innovative Functionality

As mentioned before, another option was added to the menu which utilized random numbers to pick a “special of the day” from the restaurant. This was done with two random integers chosen: one for which category the special would be from, and the other for which dish from that category. This ensured the “special” would change every time and work for every file.

```
elif menu == 's':
    x = random.randint(1, high=6)
    y = random.choice([0,2,4])
    if x == 1:
        i = appetizers
    elif x==2:
        i = entrees
    elif x==3:
        i = beverages
    elif x==4:
        i = desserts
    elif x==5:
        i = sideorders
    print('Todays Special is : ' + str(i[y]))
```

```
Select one of the menu options above
s
Todays Special is : Elote Street Corn
```

```
1 - Appetizers
2 - Entrees
3 - Beverages
4 - Desserts
5 – Side Orders
d - Display Current Cart Contents
c - Checkout
s - Todays Special
Select one of the menu options above
2
a. Japchae $15
b. Shrimp Biryani $18
c. Tamales $16
Which entree would you like?: a
Add Japchae to the cart (y/n)?: y
1 - Appetizers
2 - Entrees
3 - Beverages
4 - Desserts
5 – Side Orders
d - Display Current Cart Contents
c - Checkout
s - Todays Special
```

```
Select one of the menu options above
4
a. Songpyeon $8
b. Kheer $9
c. Churros $6
Which dessert would you like?: c
Add Churros to the cart (y/n)?: y
1 - Appetizers
2 - Entrees
3 - Beverages
4 - Desserts
5 – Side Orders
d - Display Current Cart Contents
c - Checkout
s - Todays Special
Select one of the menu options above
d

                0       $0
Japchae         1       $15
                0       $0
Churros         1       $6
                0       $0
```

```
Select one of the menu options above
c
Item            Quantity      Price

Veggie Samosas      1         $7
Japchae             1        $15
                    0         $0
Churros             1         $6
                    0         $0


Total               3                  $28
Would you like to restart and continue shopping? yes or nono
1 carts were processed
3 items bought
 28.00 total
```

```
1 - Appetizers
2 - Entrees
3 - Beverages
4 - Desserts
5 – Side Orders
d - Display Current Cart Contents
c - Checkout
s - Todays Special
Select one of the menu options above
1
a. Kimchi Pancakes $8
b. Veggie Samosas $7
c. Patatas Bravas $6
Which appetizer would you like?: b
Add Veggie Samosas to the cart (y/n)?: y
1 - Appetizers
2 - Entrees
3 - Beverages
4 - Desserts
5 – Side Orders
d - Display Current Cart Contents
c - Checkout
s - Todays Special
```

results

conclusion

- Updating this project and switching from lists to dictionaries to files gave me time to make sure things were running smoothly, and gave me lots of debugging experience
- Use ‘global’ keyword inside function definition for the variables *outside* the method you’re using *inside* a method. The method thought I was attempting to use a new, local variable inside the method, and it yelled at me for not initializing it in the method

```
aamt = 0
acost = 0
```

```
aamt += 1
acost += int(appetizers[1])
astr = appetizers[0]
```

```
def checkout():
    global aamt,acost
    a = aamt*acost
```

If not: `UnboundLocalError: local variable referenced before assignment`

Future work

This project was fun and practical, but if I had more time, I would have added a few more options on the menu that online ordering sites have, like clearing the current cart or having calories and ingredients displayed (maybe in a different font). I also would have spent more time debugging and making sure the project is able to read the information in needs from files that don’t follow the format and have multiple submenus for halal, kosher, vegan/vegetarian options