**Module 3: Critical Thinking Assignment**

Chioma Chance

CSC500

07/21/2024

# Introduction

In this assignment, I was asked to build the ShoppingCart class with specified data attributes and related methods. It is supposed to be an interactive shopping cart that allows the user to input their name along with items they want in the cart. We were even given the task of creating a menu, so the user can decide if they were ready to check out, or instead input a new quantity of their item, change the item in general, or remove the item from the cart. This is my project and how it went for me

## Step 4: Build ShoppingCart class

Source code:

# import ItemToPurchase

from Module\_4\_Portfolio\_Milestone import ItemToPurchase

# part 4: add new shoping cart class

class ShoppingCart:

    def \_\_init\_\_(self, customer\_name = 'none', current\_date = 'January 1, 2020'):

        self.customer\_name = customer\_name #attribute 1

        self.current\_date = current\_date #attribute 2

        self.cart\_items = [] #attribute 3

    def add\_item(self, item\_to\_purchase: ItemToPurchase):

        self.cart\_items.append(item\_to\_purchase)

    def remove\_item(self, item\_name: str):

        for item in self.cart\_items:

            if item.item\_name == item\_name:

                self.cart\_items.remove(item)

                return

        print('Item not found in cart. Nothing removed')

    def modify\_item(self, item\_to\_purchase: ItemToPurchase):

        for item in self.cart\_items:

            if item.item\_name == item\_to\_purchase.item\_name:

                if item\_to\_purchase.item\_price != 0.0:

                    item.item\_price = item\_to\_purchase.item\_price

                if item\_to\_purchase.item\_quantity != 0:

                    item.item\_quantity = item\_to\_purchase.item\_quantity

                return

        print('Item not found in cart. Nothing modified.')

    def get\_num\_items\_in\_cart(self):

        return sum(item.item\_quantity for item in self.cart\_items)

    def get\_cost\_of\_cart(self):

        return sum(item.item\_price \* item.item\_quantity for item in

        self.cart\_items)

    def print\_total(self):

        if not self.cart\_items:

            print('SHOPPING CART IS EMPTY')

        else:

            print(f"{self.customer\_name}'s Shopping Cart - {self.current\_date}\n")

            total\_items = self.get\_num\_items\_in\_cart()

            total\_cost = self.get\_cost\_of\_cart()

            print(f'Number of Items: {total\_items}\n')

            for item in self.cart\_items:

                print(item.print\_item\_cost())

            print(f'\nTotal: ${total\_cost:.2f}')

    def print\_descriptions(self):

        print(f'{self.customer\_name} - {self.current\_date}\n')

        print('Item Descriptions\n')

        for item in self.cart\_items:

            description = input(f'Enter description for {item.item\_name}:')

            print(f'{item.item\_name}: {description}')

# Part 5 & 6: Build Menu and Implement Output

Source code:

# part 5

def print\_menu(shopping\_cart: ShoppingCart):

    print('\nMENU')

    print('q - Quit\n')

    print('a - Add item to cart\n')

    print('r - Remove item from cart\n')

    print('c - Change item quantity\n')

    print('i - Output items\' descriptions\n')

    print('o - Output shopping cart\n')

    print('Choose an option: ', end='')

# def main: implement instructions for menu

def main():

    #user input for name

    customer\_name = input('Please enter your name: ')

    cart = ShoppingCart(customer\_name)

    while True:

        print\_menu()

        # make sure there are no proceeding spaces and letters are lowercase

        choice = input('Choose an option').strip().lower()

        if choice == 'a':

            name = input('Enter item name:\n')

            price = float(input('Enter the item price:\n'))

            quantity = int(input('Enter the item quantity:\n'))

            cart.add\_item(ItemToPurchase(name, price, quantity))

            print('Item added')

        elif choice == 'r':

            # remove item

            name = input('Enter item name to remove: ')

            cart.remove\_item(name)

            print('Item removed')

        elif choice == 'c':

            # change item quantity

            name = input('Enter name of item you want to modify: ')

            new\_quantity = int(input('Enter new quantity: '))

            item\_to\_modify = ItemToPurchase(name, '', 0.0, new\_quantity)

            cart.modify\_item(item\_to\_modify)

            print('Item quantity modified')

        elif choice == 'i':

            cart.print\_descriptions()

        elif choice == 'o':

            cart.print\_total()

        elif choice == 'q':

            # quit program

            print('Thank you for using the virtual shopping cart, enjoy :)')

            break

        else:

            print('Invalid option, plaese try again.')

Screenshots:

A screenshot of a computer screen

Description automatically generatedA computer screen shot of a program

Description automatically generatedA computer screen shot of a black screen

Description automatically generated

# **GIT repository link**

<https://github.com/Ch1T1me/CSC500.git>

# **Challenges**

This assignment presented numerous challenges that tested my understanding and patience. Here are some of the specific difficulties I encountered:

1. **Defining print\_descriptions()** Setting up the print\_descriptions() method was particularly challenging because I wasn’t sure how to handle item descriptions that users would input dynamically. For example, if a user added marshmallows to their cart, how could I have a pre-existing description ready for that? I realized there wasn’t a built-in function to generate descriptions automatically based on user input. Therefore, I decided to prompt the user to enter descriptions themselves using input statements. This solution seemed logical, but it was still a tricky problem to solve.
2. **Setting Up print\_menu()** The print\_menu() method was another stumbling block. I struggled with understanding how to set it up and integrate it properly within the main() function. Despite watching numerous videos and examining other code examples, I remained confused about the instructions to call print\_menu() within main(). This confusion set me back significantly. Thankfully, Dr. Evans’ video provided much-needed clarity, allowing me to move forward.
3. **Implementing main()** Developing the main() function was tough, particularly when setting the conditions for each menu item. The most challenging part was handling the choice == 'c' case. Even now, I don’t fully understand the implementation I used, which was frustrating.
4. **Debugging Attribute Access** While seeking help for my code, I noticed that item.item\_price, item.item\_name, and item.item\_quantity were not highlighting yellow, indicating an issue. I intended to import these attributes from the ItemToPurchase class but clearly didn’t do it correctly. This persistent problem left me puzzled, and I’m still unsure where I went wrong.
5. **Overwhelmed by Code from Module 4** Initially, I had the entire code from Module 4 in my project, which was overwhelming. Remembering the textbook’s guidance on importing files, I simplified my script with from Module\_4\_Portfolio\_Milestone import ItemToPurchase. This helped reduce the complexity, but it was still a challenging start.
6. **Overall Code Functionality** Despite my best efforts, my code doesn’t work as intended. I thought I followed all the instructions correctly, but that wasn’t the case. I would greatly appreciate feedback on how to correct my mistakes before the next portfolio project. I really exerted myself on this assignment and feel disheartened that it didn’t work out as expected.

In conclusion, this project was incredibly challenging, but I learned a lot from the experience. One of the most confusing aspects was that my code only outputted the values from Module 4. I expected it to integrate and handle the new functionalities correctly, but instead, it just echoed the previous module’s outputs. This added to my frustration and made it difficult to troubleshoot and understand where I went wrong. Any guidance on improving my approach and understanding would be invaluable.

**References**

CSC500 Principles of Programming. (n.d.). In ZyBooks. Retrieved from <https://learn.zybooks.com/zybook/CSUGLOBALCSC500MASTER1>

Stack Overflow. (2013, November 13). Creating a menu in Python. Retrieved from <https://stackoverflow.com/questions/19964603/creating-a-menu-in-python>

Tech With Tim. (2020, March 11). Python Beginner Tutorial: Creating a Menu in Python. [Video]. YouTube. <https://youtu.be/huUg6wDUsv4?si=jGsLRfOFbZUhN_zV>

Pluralsight. (n.d.). Manipulating lists and dictionaries in Python. Retrieved from <https://www.pluralsight.com/resources/blog/guides/manipulating-lists-dictionaries-python>

Python Course. (n.d.). Dictionaries. Retrieved from <https://python-course.eu/python-tutorial/dictionaries.php>

Python Basics. (n.d.). Python Lists. Retrieved from <https://pythonbasics.org/list/>

Python Basics. (n.d.). Python Classes. Retrieved from <https://pythonbasics.org/class/>

OpenAI. (2023). ChatGPT (July 2023 version) [Large language model]. Retrieved from <https://www.openai.com/chatgpt>