**Module 8: Portfolio Milestone**

Chioma Chance

CSC500

08/03/2024

# Introduction

Module 8 Portfolio Project was the collection of all the portfolios in one. We started with an ItemToPurchase class then moved on to the ShoppingCart class. The purpose of this project was to make sure each of the classes are working properly for user engagement. I updated everything that needed fixing, and made sure each item in my menu worked properly.

## **Source Code**

# Online Shopping Cart

class ItemToPurchase:

    def \_\_init\_\_(self, name = 'none', price = 0.0, quantity = 0):

        self.item\_name = name

        self.item\_price = price

        self.item\_quantity = quantity

    def print\_item\_cost(self):

        total\_cost = self.item\_price \* self.item\_quantity

        return f'{self.item\_name} {self.item\_quantity} @ ${self.item\_price} = ${total\_cost:.2f}'

# 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 = getattr(item, 'description', 'No description available')

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

# 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():

    print('Welcome to the online Shopping Cart\n')

    #user input for name

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

    current\_date = input('Please enter today\'s date:\n')

    cart = ShoppingCart(customer\_name, current\_date)

    print(f'Customers name: {customer\_name}')

    print(f'Today\'s date: {current\_date}')

    while True:

        print\_menu(cart)

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

        choice = input().strip().lower()

        if choice == 'a':

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

            description = input('Enter product description:\n')

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

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

            item = ItemToPurchase(name, price, quantity)

            item.description = description

            cart.add\_item(item)

            print(f'Added {name} to cart.')

        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 to {new\_quantity}')

        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.')

# call main function to action

if \_\_name\_\_ == '\_\_main\_\_':

    main()

# **Pseudocode**

Class ItemToPurchase:

Function \_\_init\_\_(name = 'none', price = 0.0, quantity = 0):

Set self.item\_name to name

Set self.item\_price to price

Set self.item\_quantity to quantity

Function print\_item\_cost():

Set total\_cost to self.item\_price \* self.item\_quantity

Return formatted string of item\_name, item\_quantity, item\_price, and total\_cost

Class ShoppingCart:

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

Set self.customer\_name to customer\_name

Set self.current\_date to current\_date

Set self.cart\_items to empty list

Function add\_item(item\_to\_purchase):

Append item\_to\_purchase to self.cart\_items

Function remove\_item(item\_name):

For each item in self.cart\_items:

If item.item\_name equals item\_name:

Remove item from self.cart\_items

Return

Print 'Item not found in cart. Nothing removed'

Function modify\_item(item\_to\_purchase):

For each item in self.cart\_items:

If item.item\_name equals item\_to\_purchase.item\_name:

If item\_to\_purchase.item\_price is not 0.0:

Set item.item\_price to item\_to\_purchase.item\_price

If item\_to\_purchase.item\_quantity is not 0:

Set item.item\_quantity to item\_to\_purchase.item\_quantity

Return

Print 'Item not found in cart. Nothing modified.'

Function get\_num\_items\_in\_cart():

Return sum of item.item\_quantity for each item in self.cart\_items

Function get\_cost\_of\_cart():

Return sum of item.item\_price \* item.item\_quantity for each item in self.cart\_items

Function print\_total():

If self.cart\_items is empty:

Print 'SHOPPING CART IS EMPTY'

Else:

Print customer\_name and current\_date

Set total\_items to result of get\_num\_items\_in\_cart

Set total\_cost to result of get\_cost\_of\_cart

Print 'Number of Items:', total\_items

For each item in self.cart\_items:

Print item.print\_item\_cost()

Print 'Total:', total\_cost

Function print\_descriptions():

Print customer\_name and current\_date

Print 'Item Descriptions'

For each item in self.cart\_items:

Set description to item.description if exists else 'No description available'

Print item.item\_name and description

Function print\_menu(shopping\_cart):

Print menu options

Function main():

Print welcome message

Input customer\_name

Input current\_date

Create ShoppingCart object cart with customer\_name and current\_date

Print customer\_name and current\_date

While True:

Call print\_menu(cart)

Input choice and convert to lowercase

If choice equals 'a':

Input item name

Input item description

Input item price and convert to float

Input item quantity and convert to integer

Create ItemToPurchase object item with name, price, and quantity

Set item.description to description

Call cart.add\_item(item)

Print 'Added {name} to cart.'

Else If choice equals 'r':

Input item name to remove

Call cart.remove\_item(name)

Print 'Item removed'

Else If choice equals 'c':

Input item name to modify

Input new quantity and convert to integer

Create ItemToPurchase object item\_to\_modify with name, 0.0, and new\_quantity

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

Print 'Item quantity modified to {new\_quantity}'

Else If choice equals 'i':

Call cart.print\_descriptions()

Else If choice equals 'o':

Call cart.print\_total()

Else If choice equals 'q':

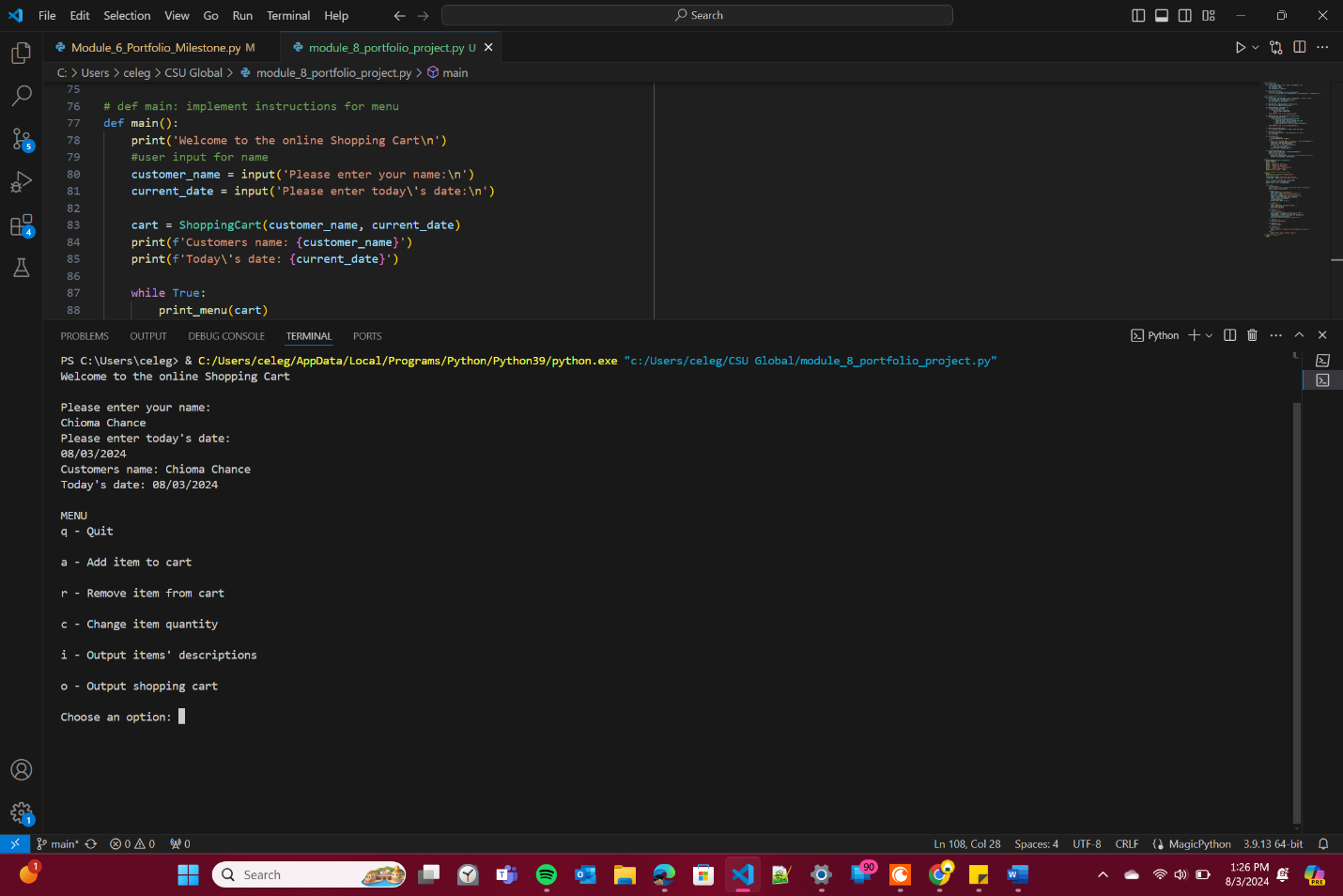
Print 'Thank you for using the virtual shopping cart, enjoy :)'

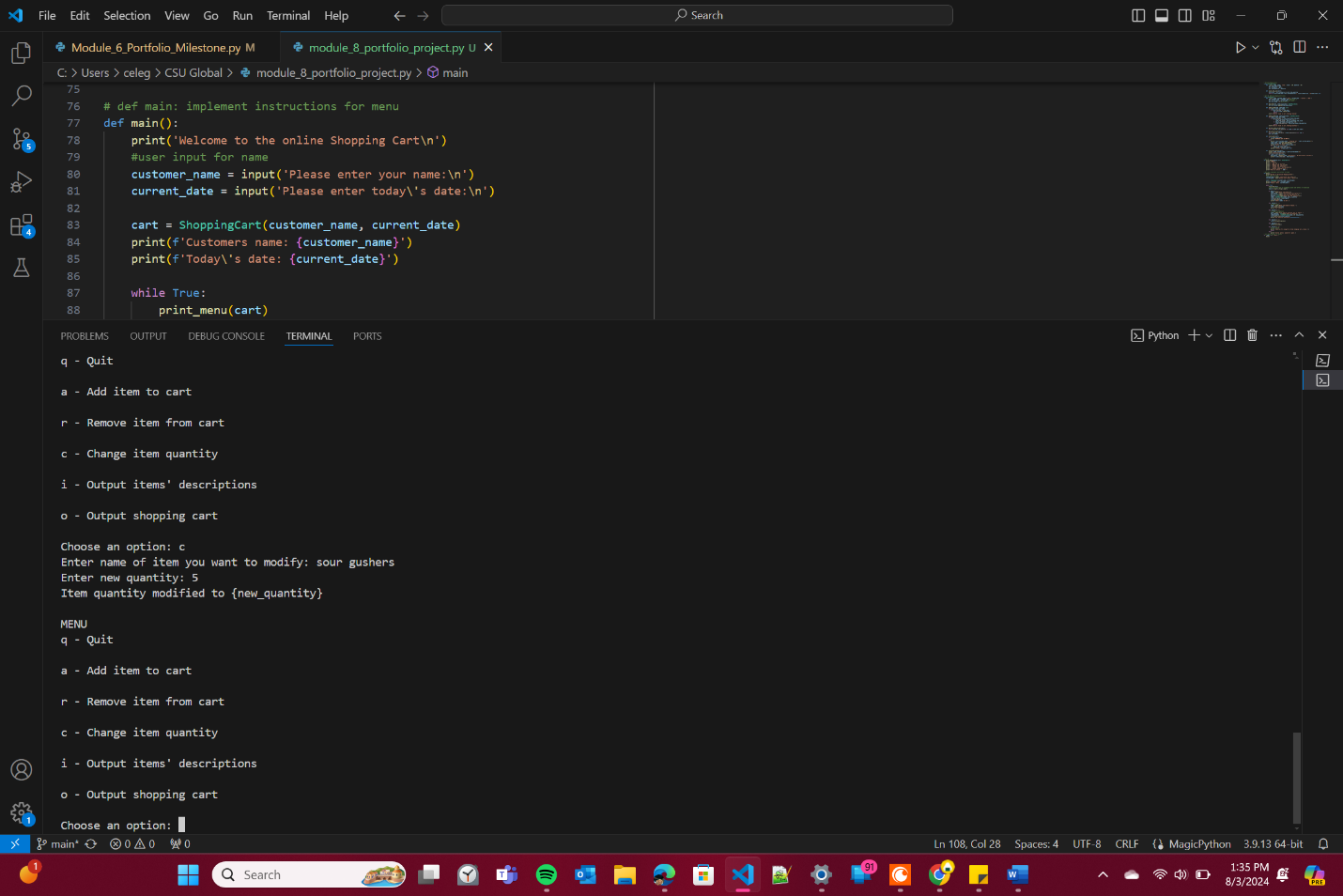
Break

Else:

Print 'Invalid option, please try again.'

Call main function if \_\_name\_\_ equals '\_\_main\_\_':

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**GIT repository link**

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

# **Challenges**

I didn't face any challenges in this project, thanks to all the hard work done in Module 6. During Module 6, I encountered numerous difficulties but, with diligence and valuable feedback from you, Dr. Evans, I was able to overcome them. This week was mainly about writing up my findings and reflecting on the learning process. I've gained so many new skills throughout this course and these projects, and I truly appreciate the guidance and support provided. Thank you.

**References**

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

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

 "9. Classes." *Python Documentation*, Python Software Foundation, <https://docs.python.org/3/tutorial/classes.html>. Accessed 3 Aug. 2024.

 "Python String." *GeeksforGeeks*, <https://www.geeksforgeeks.org/python-string/>. Accessed 3 Aug. 2024.