**Portfolio Milestone 6 Online Shopping Cart-**

References:

[1] Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media. ISBN: 978-1449355739.

[2] Python Software Foundation. (2023). Python 3.9.1 documentation - Classes. Retrieved from https://docs.python.org/3/tutorial/classes.html

[3] W3Schools. (2023). Python Classes/Objects. Retrieved from https://www.w3schools.com/python/python\_classes.asp

[4] https://learn.zybooks.com/zybook/CSUGLOBALCSC500MASTER1/chapter/8

[5] https://learn.zybooks.com/zybook/CSUGLOBALCSC500MASTER1/chapter/6

Few interesting learnings while preparing for and writing code for this project-

Lists in Python are dynamic arrays, meaning they can grow or shrink in size as needed. This makes them ideal for situations where you don't know the number of elements in advance, such as when constructing an object that will hold a collection of items (like items in a shopping cart).

We can initialize a list in a constructor with parameters to provide initial values for the object's attributes. If the list should be empty at the start, we can initialize it as an empty list using [] ( like the cart\_items )

I also learned about the main() function in Python. In Python, the main function isn't a built-in concept like it is in some other programming languages, such as C or Java. However, it's a convention that's often used in Python scripts to denote the entry point of a program. The main function is typically used to organize the flow of execution in a script, making it more structured and readable.

The if \_\_name\_\_ == "\_\_main\_\_": check is used to determine if the script is being run directly or being imported as a module into another script. When the script is run directly (for example, by executing python script.py on the command line), the \_\_name\_\_ variable is set to "\_\_main\_\_", and the code inside the if block will execute, calling the main function.

**Pseudocode-**

Class ItemToPurchase:

Declare item\_name as String

Declare item\_price as Float

Declare item\_quantity as Int

Method \_\_init\_\_(item\_name="none", item\_price=0, item\_quantity=0):

Set self.item\_name to item\_name

Set self.item\_price to item\_price

Set self.item\_quantity to item\_quantity

Method print\_item\_cost():

Calculate total\_cost as item\_price \* item\_quantity

Print item\_name, item\_quantity, item\_price, and total\_cost

Class ShoppingCart:

Declare customer\_name as String

Declare current\_date as String

Declare cart\_items as List of ItemToPurchase

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

Set self.customer\_name to customer\_name

Set self.current\_date to current\_date

Initialize self.cart\_items as an empty list

Method add\_item(ItemToPurchase):

Append ItemToPurchase to cart\_items

Method remove\_item(item\_name):

For each item in cart\_items:

If item's name matches item\_name:

Remove item from cart\_items

Return

Print "Item not found in cart. Nothing removed."

Method modify\_item(ItemToPurchase):

For each item in cart\_items:

If item's name matches ItemToPurchase's name:

Update item's price and quantity from ItemToPurchase

Return

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

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

Declare num\_items as Int and set to 0

For each item in cart\_items:

Add item's quantity to num\_items

Return num\_items

Method get\_cost\_of\_cart():

Declare total\_cost as Float and set to 0

For each item in cart\_items:

Add item's price \* quantity to total\_cost

Return total\_cost

Method print\_total():

Print customer\_name, current\_date, and number of items in cart

If cart\_items is empty:

Print "SHOPPING CART IS EMPTY"

Else:

For each item in cart\_items:

Call item's print\_item\_cost method

Print total cost of cart

Method print\_descriptions():

Print customer\_name, current\_date, and "Item Descriptions"

For each item in cart\_items:

Print item's name and quantity

Method print\_menu(ShoppingCart):

Declare command as String

While command is not "q":

Print menu options

Get user input for command

If command is "a":

Prompt for item\_name, item\_price, item\_quantity

Call ShoppingCart's add\_item method

Else If command is "r":

Prompt for item\_name

Call ShoppingCart's remove\_item method

Else If command is "c":

Implement change item quantity functionality

Else If command is "i":

Call ShoppingCart's print\_descriptions method

Else If command is "o":

Call ShoppingCart's print\_total method

Else If command is "q":

Print "Quitting..."

Else:

Print "Invalid option, please try again."

Method main():

Create ShoppingCart instance with default or user-provided values

Call print\_menu with ShoppingCart instance

Call main() if script is run directly

**Source code –**

Below are the screen shots of the source code

I have added the link to my .py file and uploaded as well.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Screen Shots of the Executed code with errors and fixes-**

A screenshot of a computer

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer program

Description automatically generated

Successful execution of all the methods “options” , in the MENU

A screenshot of a computer screen

Description automatically generatedA screenshot of a computer

Description automatically generated