CSC500

Module 4 Portfolio Milestone

GIT Repository: <https://github.com/focomapper/CSC500>

Introduction:

The module 4 Portfolio Milestone assignment consisted of creating an items to purchase class. This class required three variables – item name, price an quantity. The class also needed a default constructor (no arguments) and initialized the three variables and a method to print the variable values along with a sum of the cost for the item. The Main code was required to get input from a user for two items, instantiate objects from the class for both items, and print a summary using the class method along with a sum of the total price for both items. While I did not have any difficulty developing this code, I did enjoy the process and spent time refining the code to be as concise as I could get it.

Pseudo code-

1. Create Item To Purchase class
   1. Add constructor and initialize variables.
   2. Develop method to print values of variables.
2. Develop main code
   1. For loop to run twice
   2. Initialize list to hold objects from class
   3. Instantiate class into list
   4. Get user input for each two items (validate input)
   5. Populate object variables with user input
   6. Loop through list, run class print method, and print summary of items

Source code-

# Create Item To Purchase class  
class ItemToPurchase:  
 def \_\_init\_\_(self):  
 # Initialize variables  
 self.item\_name = "none"  
 self.item\_price = float(0)  
 self.item\_quantity = 0  
  
 # Print item method  
 def print\_item\_cost(self):  
 total\_price = self.item\_price \* self.item\_quantity  
 print(self.item\_name + ' ' + str(self.item\_quantity) + ' @ ${:,.2f}'.format(self.item\_price) + '= ${:,.2f}'.format(total\_price))

# Main code  
  
# Initialize list of items  
list\_items = []  
for j in range(2): # Loop through 2 times, can change later or modify to prompt user to end input  
 # Instantiate class and add to list  
 list\_items.append(ItemToPurchase())  
 list\_items[j].item\_name = str(input("Enter the item name: "))  
 try: # Ensure price is number, exit if not  
 list\_items[j].item\_price = float(input("Enter the item price: "))  
 except ValueError:  
 print("Your input for price was not a number, run program again and enter valid price")  
 exit()  
 try: # Ensure quantity is number, exit if not  
 list\_items[j].item\_quantity = int(input("Enter the item quantity: "))  
 except ValueError:  
 print("Your input for quantity was not a number, run program again and enter valid quantity")  
 exit()  
  
# Print required output  
total\_cost = 0  
print("Total Cost")  
i = 0  
while i < len(list\_items): # Loop through list of item objects, run print method, and calculate/print total cost  
 list\_items[i].print\_item\_cost()  
 total\_cost += list\_items[i].item\_price \* list\_items[i].item\_quantity  
 i += 1  
print('Total: ' + '${:,.2f}'.format(total\_cost))

Screenshot of code executing-

A screenshot of a computer program

Description automatically generated