

Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

class ItemToPurchase:

def \_\_init\_\_(self, item\_name="none", item\_description="none", item\_price=0, item\_quantity=0):

self.item\_name = item\_name

self.item\_description = item\_description

self.item\_price = item\_price

self.item\_quantity = item\_quantity

class ShoppingCart:

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

self.customer\_name = customer\_name

self.current\_date = current\_date

self.cart\_items = []

def add\_item(self, item):

self.cart\_items.append(item)

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

item\_found = False

for item in self.cart\_items:

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

self.cart\_items.remove(item)

item\_found = True

break

if not item\_found:

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

def modify\_item(self, modified\_item):

item\_found = False

for item in self.cart\_items:

if item.item\_name == modified\_item.item\_name:

item\_found = True

if modified\_item.item\_description != "none":

item.item\_description = modified\_item.item\_description

if modified\_item.item\_price != 0:

item.item\_price = modified\_item.item\_price

if modified\_item.item\_quantity != 0:

item.item\_quantity = modified\_item.item\_quantity

break

if not item\_found:

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

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

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

return total\_quantity

def get\_cost\_of\_cart(self):

total\_cost = sum(item.item\_price \* item.item\_quantity for item in self.cart\_items)

return total\_cost

def print\_total(self):

print(f"OUTPUT SHOPPING CART")

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

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

print(f"Number of Items: {num\_items}")

if num\_items == 0:

print("SHOPPING CART IS EMPTY")

else:

for item in self.cart\_items:

item\_total = item.item\_price \* item.item\_quantity

print(f"{item.item\_name} {item.item\_quantity} @ ${item.item\_price} = ${item\_total}")

print(f"Total: ${self.get\_cost\_of\_cart()}")

def print\_descriptions(self):

print(f"OUTPUT ITEMS' DESCRIPTIONS")

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

print("Item Descriptions")

for item in self.cart\_items:

print(f"{item.item\_name}: {item.item\_description}")

def print\_menu(cart):

menu\_text = (

"\nMENU\n"

"a - Add item to cart\n"

"r - Remove item from cart\n"

"c - Change item quantity\n"

"i - Output items' descriptions\n"

"o - Output shopping cart\n"

"q - Quit\n"

"Choose an option:"

)

while True:

print(menu\_text)

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

if choice == 'a':

item\_name = input("Enter the item name: ")

item\_description = input("Enter the item description: ")

item\_price = int(input("Enter the item price: "))

item\_quantity = int(input("Enter the item quantity: "))

new\_item = ItemToPurchase(item\_name, item\_description, item\_price, item\_quantity)

cart.add\_item(new\_item)

elif choice == 'r':

item\_name = input("Enter name of item to remove: ")

cart.remove\_item(item\_name)

elif choice == 'c':

item\_name = input("Enter the item name: ")

item\_description = input("Enter new description (or 'none' to keep unchanged): ")

item\_price = int(input("Enter new price (or 0 to keep unchanged): "))

item\_quantity = int(input("Enter new quantity (or 0 to keep unchanged): "))

modified\_item = ItemToPurchase(item\_name, item\_description, item\_price, item\_quantity)

cart.modify\_item(modified\_item)

elif choice == 'i':

cart.print\_descriptions()

elif choice == 'o':

cart.print\_total()

elif choice == 'q':

break

else:

print("Invalid option. Please try again.")

def main():

customer\_name = input("Enter customer's name: ")

current\_date = input("Enter today's date: ")

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

print\_menu(cart)

if \_\_name\_\_ == "\_\_main\_\_":

main()

