Brandon Everett

CSC500-2

Portfolio Project

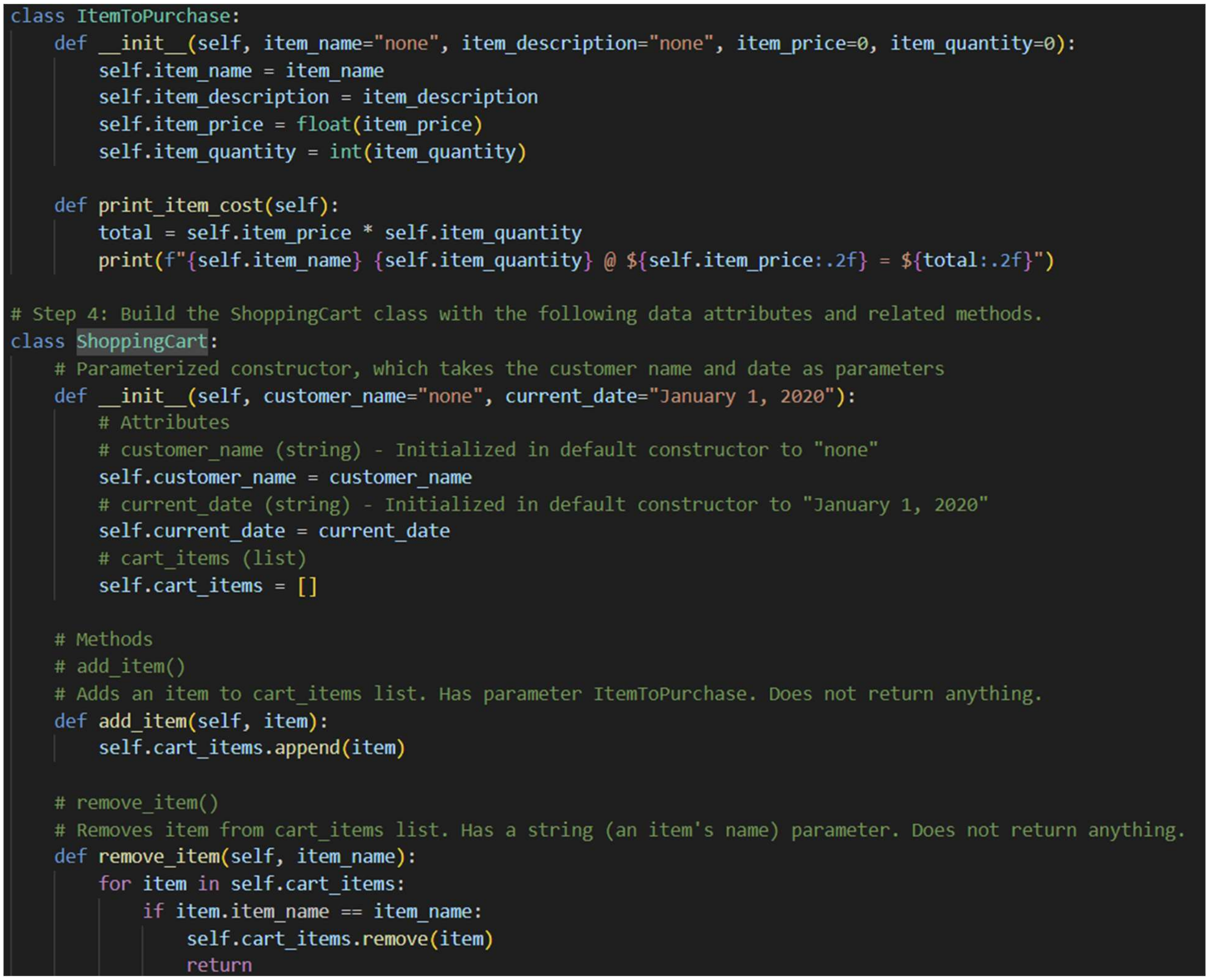
11/3/2024

Online Shopping Cart

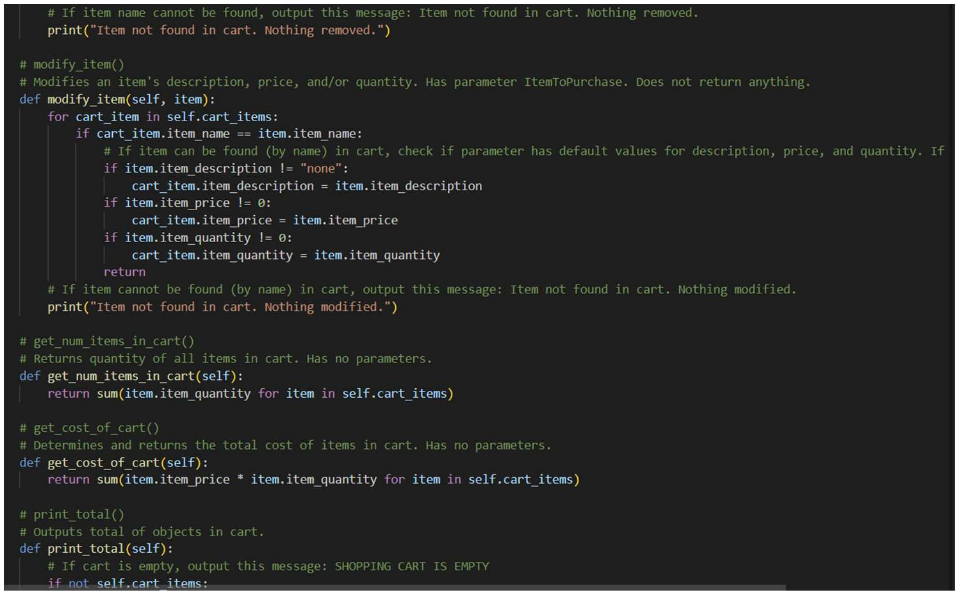
For the final Portfolio Project, I completed an enhanced version of the online shopping cart program that demonstrates the culmination of everything I've learned throughout this course. Interestingly, I had already implemented most of the required functionality during Module 6, not realizing I was working ahead on what would become the final project. The only significant addition needed for the final submission was implementing the display of customer name and today's date.

The project consists of two main classes: ItemToPurchase and ShoppingCart. The ItemToPurchase class stores information about individual items including name, description, price, and quantity, while calculating and displaying costs. The ShoppingCart class manages the collection of items, implementing methods for adding, removing, and modifying items, as well as calculating totals and displaying summaries. The program features a user-friendly menu system that allows customers to manage their cart through various options while maintaining customer information and purchase dates.

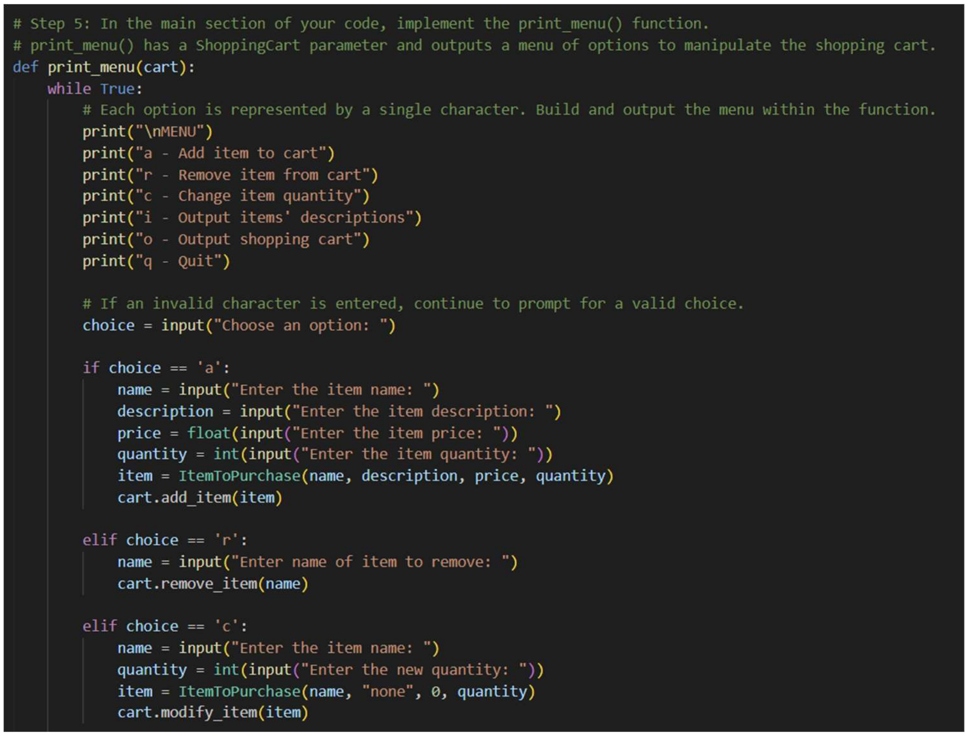
This course has given me a good starting point for understanding Python programming. While I still have a lot to learn, I've come a long way from struggling with basic concepts in the first few weeks. Each module helped me understand a little more, and by the time I got to this final project, things started making more sense. I'm proud of how far I've come and looking forward to continuing to improve my programming skills. Thank you!



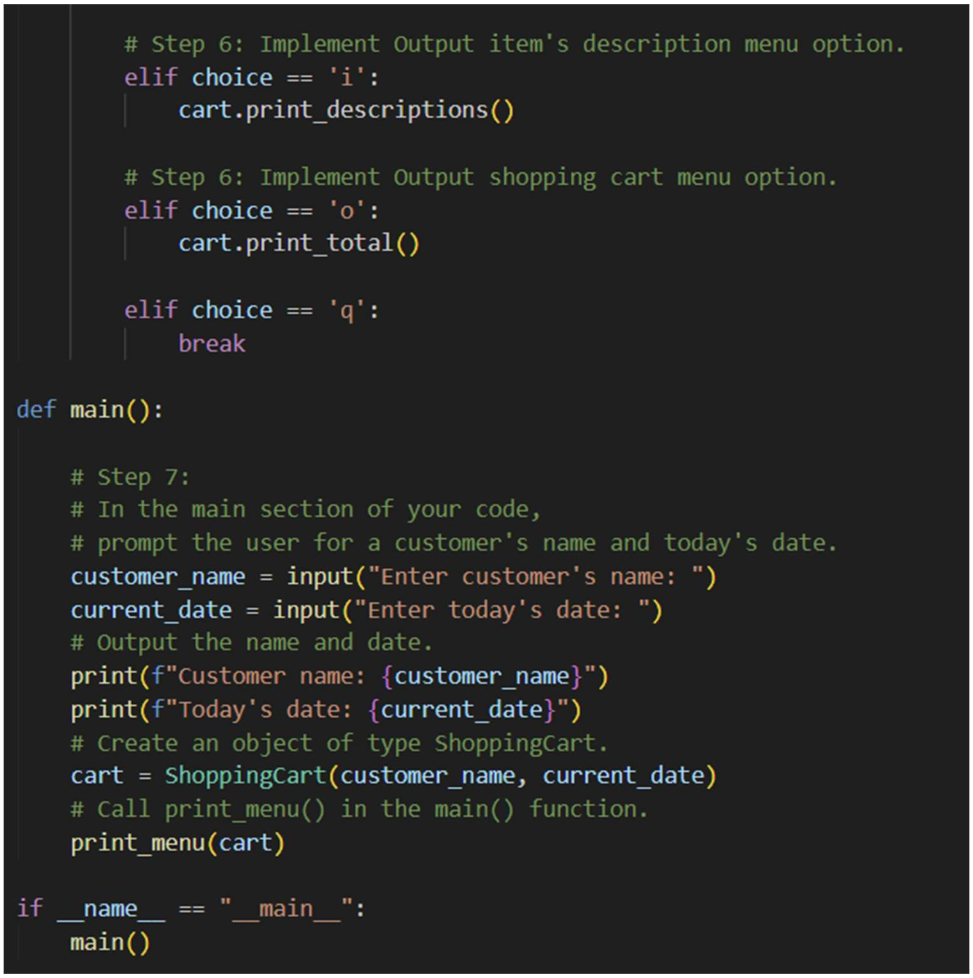
```python
class ItemToPurchase:
    def __init__(self, item_name="none", item_description="none", item_price=0, item_quantity=0):
        self.item_name = item_name
        self.item_description = item_description
        self.item_price = float(item_price)
        self.item_quantity = int(item_quantity)

    def print_item_cost(self):
        total = self.item_price * self.item_quantity
        print(f"{self.item_name} {self.item_quantity} @ ${self.item_price:.2f} = ${total:.2f}")

# Step 4: Build the ShoppingCart class with the following data attributes and related methods.
class ShoppingCart:
    # Parameterized constructor, which takes the customer name and date as parameters
    def __init__(self, customer_name="none", current_date="January 1, 2020"):
        # Attributes
        # customer_name (string) - Initialized in default constructor to "none"
        self.customer_name = customer_name
        # current_date (string) - Initialized in default constructor to "January 1, 2020"
        self.current_date = current_date
        # cart_items (list)
        self.cart_items = []

    # Methods
    # add_item()
    # Adds an item to cart_items list. Has parameter ItemToPurchase. Does not return anything.
    def add_item(self, item):
        self.cart_items.append(item)

    # remove_item()
    # Removes item from cart_items list. Has a string (an item's name) parameter. Does not return anything.
    def remove_item(self, item_name):
        for item in self.cart_items:
            if item.item_name == item_name:
                self.cart_items.remove(item)
                return
```



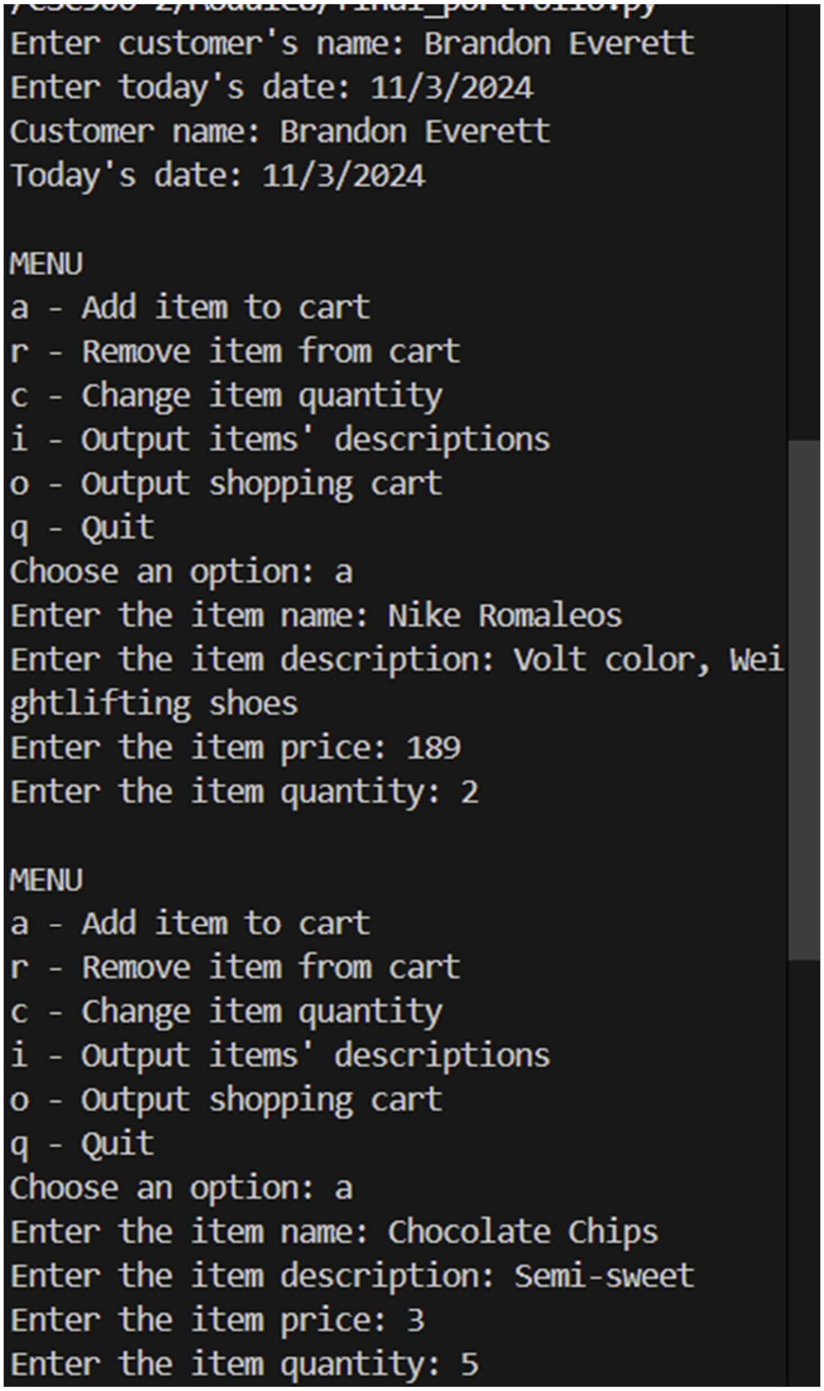
```python
        # If item name cannot be found, output this message: Item not found in cart. Nothing removed.
        print("Item not found in cart. Nothing removed.")

    # modify_item()
    # Modifies an item's description, price, and/or quantity. Has parameter ItemToPurchase. Does not return anything.
    def modify_item(self, item):
        for cart_item in self.cart_items:
            if cart_item.item_name == item.item_name:
                # If item can be found (by name) in cart, check if parameter has default values for description, price, and quantity. If
                if item.item_description != "none":
                    cart_item.item_description = item.item_description
                if item.item_price != 0:
                    cart_item.item_price = item.item_price
                if item.item_quantity != 0:
                    cart_item.item_quantity = item.item_quantity
                return
        # If item cannot be found (by name) in cart, output this message: Item not found in cart. Nothing modified.
        print("Item not found in cart. Nothing modified.")

    # get_num_items_in_cart()
    # Returns quantity of all items in cart. Has no parameters.
    def get_num_items_in_cart(self):
        return sum(item.item_quantity for item in self.cart_items)

    # get_cost_of_cart()
    # Determines and returns the total cost of items in cart. Has no parameters.
    def get_cost_of_cart(self):
        return sum(item.item_price * item.item_quantity for item in self.cart_items)

    # print_total()
    # Outputs total of objects in cart.
    def print_total(self):
        # If cart is empty, output this message: SHOPPING CART IS EMPTY
        if not self.cart_items:
```



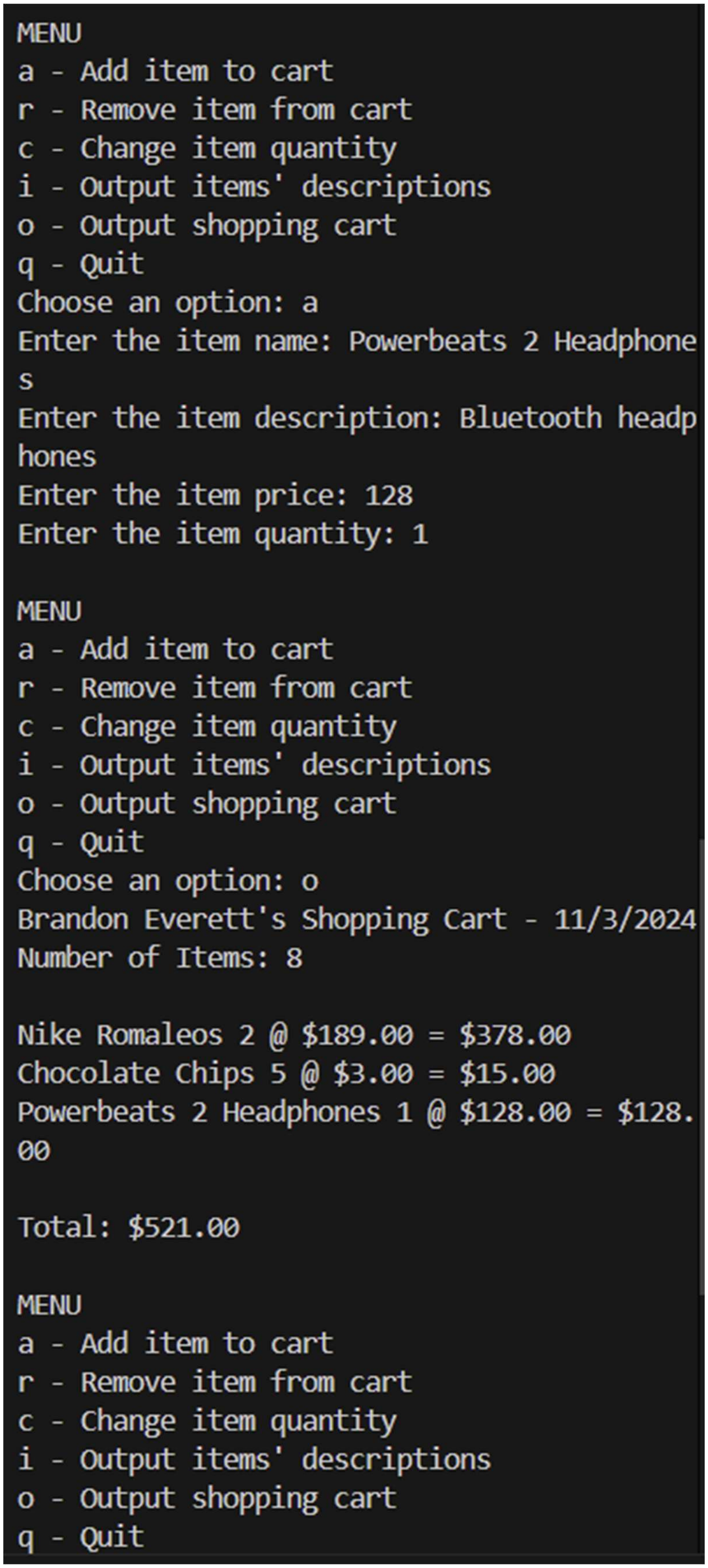
```python
# Step 5: In the main section of your code, implement the print_menu() function.
# print_menu() has a ShoppingCart parameter and outputs a menu of options to manipulate the shopping cart.
def print_menu(cart):
    while True:
        # Each option is represented by a single character. Build and output the menu within the function.
        print("\nMENU")
        print("a - Add item to cart")
        print("r - Remove item from cart")
        print("c - Change item quantity")
        print("i - Output items' descriptions")
        print("o - Output shopping cart")
        print("q - Quit")

        # If an invalid character is entered, continue to prompt for a valid choice.
        choice = input("Choose an option: ")

        if choice == 'a':
            name = input("Enter the item name: ")
            description = input("Enter the item description: ")
            price = float(input("Enter the item price: "))
            quantity = int(input("Enter the item quantity: "))
            item = ItemToPurchase(name, description, price, quantity)
            cart.add_item(item)

        elif choice == 'r':
            name = input("Enter name of item to remove: ")
            cart.remove_item(name)

        elif choice == 'c':
            name = input("Enter the item name: ")
            quantity = int(input("Enter the new quantity: "))
            item = ItemToPurchase(name, "none", 0, quantity)
            cart.modify_item(item)
```



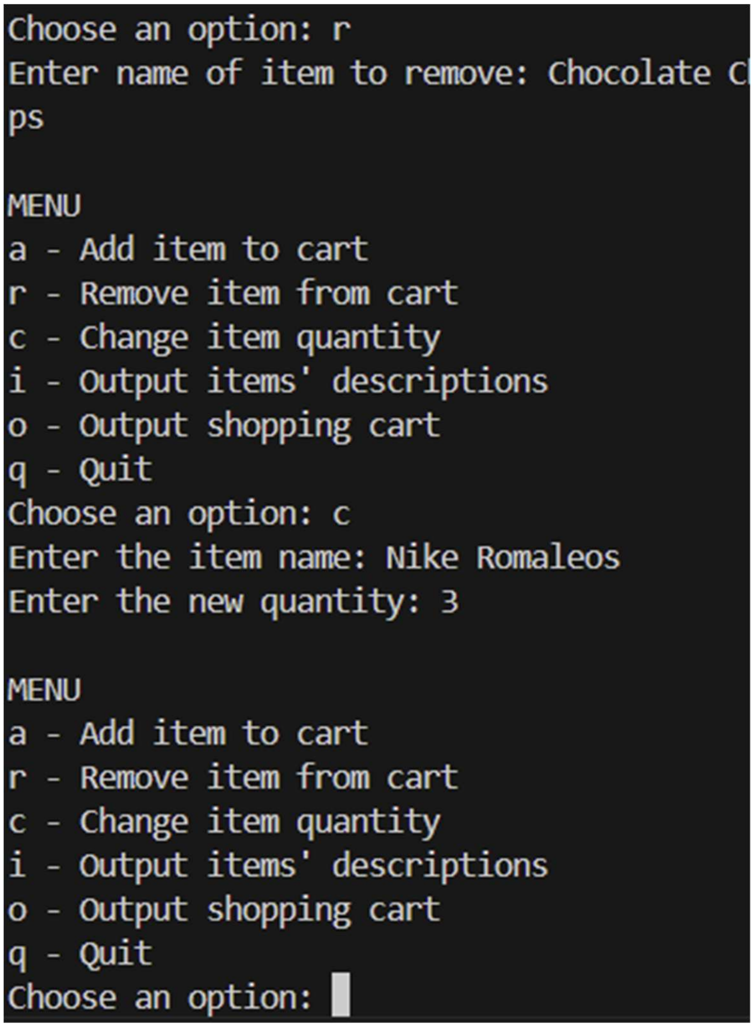
```python
        # Step 6: Implement Output item's description menu option.
        elif choice == 'i':
            cart.print_descriptions()

        # Step 6: Implement Output shopping cart menu option.
        elif choice == 'o':
            cart.print_total()

        elif choice == 'q':
            break

def main():

    # Step 7:
    # In the main section of your code,
    # prompt the user for a customer's name and today's date.
    customer_name = input("Enter customer's name: ")
    current_date = input("Enter today's date: ")
    # Output the name and date.
    print(f"Customer name: {customer_name}")
    print(f"Today's date: {current_date}")
    # Create an object of type ShoppingCart.
    cart = ShoppingCart(customer_name, current_date)
    # Call print_menu() in the main() function.
    print_menu(cart)

if __name__ == "__main__":
    main()
```



```
/CSC500-2/Modules/final_portfolio.py
Enter customer's name: Brandon Everett
Enter today's date: 11/3/2024
Customer name: Brandon Everett
Today's date: 11/3/2024

MENU
a - Add item to cart
r - Remove item from cart
c - Change item quantity
i - Output items' descriptions
o - Output shopping cart
q - Quit
Choose an option: a
Enter the item name: Nike Romaleos
Enter the item description: Volt color, Wei
ghtlifting shoes
Enter the item price: 189
Enter the item quantity: 2

MENU
a - Add item to cart
r - Remove item from cart
c - Change item quantity
i - Output items' descriptions
o - Output shopping cart
q - Quit
Choose an option: a
Enter the item name: Chocolate Chips
Enter the item description: Semi-sweet
Enter the item price: 3
Enter the item quantity: 5
```



```
MENU
a - Add item to cart
r - Remove item from cart
c - Change item quantity
i - Output items' descriptions
o - Output shopping cart
q - Quit
Choose an option: a
Enter the item name: Powerbeats 2 Headphone
s
Enter the item description: Bluetooth headp
hones
Enter the item price: 128
Enter the item quantity: 1

MENU
a - Add item to cart
r - Remove item from cart
c - Change item quantity
i - Output items' descriptions
o - Output shopping cart
q - Quit
Choose an option: o
Brandon Everett's Shopping Cart - 11/3/2024
Number of Items: 8

Nike Romaleos 2 @ $189.00 = $378.00
Chocolate Chips 5 @ $3.00 = $15.00
Powerbeats 2 Headphones 1 @ $128.00 = $128.
00

Total: $521.00

MENU
a - Add item to cart
r - Remove item from cart
c - Change item quantity
i - Output items' descriptions
o - Output shopping cart
q - Quit
```



```
Choose an option: r
Enter name of item to remove: Chocolate C
ps

MENU
a - Add item to cart
r - Remove item from cart
c - Change item quantity
i - Output items' descriptions
o - Output shopping cart
q - Quit
Choose an option: c
Enter the item name: Nike Romaleos
Enter the new quantity: 3

MENU
a - Add item to cart
r - Remove item from cart
c - Change item quantity
i - Output items' descriptions
o - Output shopping cart
q - Quit
Choose an option:
```

Sources:

GeeksforGeeks. "Python Classes and Objects." GeeksforGeeks, 15 Aug 2024, www.geeksforgeeks.org/python-classes-and-objects/. Accessed 20 Oct. 2024.

GeeksforGeeks. "Python List Comprehension." GeeksforGeeks, 30 July 2024, www.geeksforgeeks.org/python-list-comprehension/. Accessed 20 Oct. 2024.

GeeksforGeeks. "Object Oriented Programming in Python." GeeksforGeeks, 5 Sept 2024, www.geeksforgeeks.org/object-oriented-programming-in-python/. Accessed 20 Oct. 2024.