Module 4 Portfolio Project

Alex Zelmanowicz

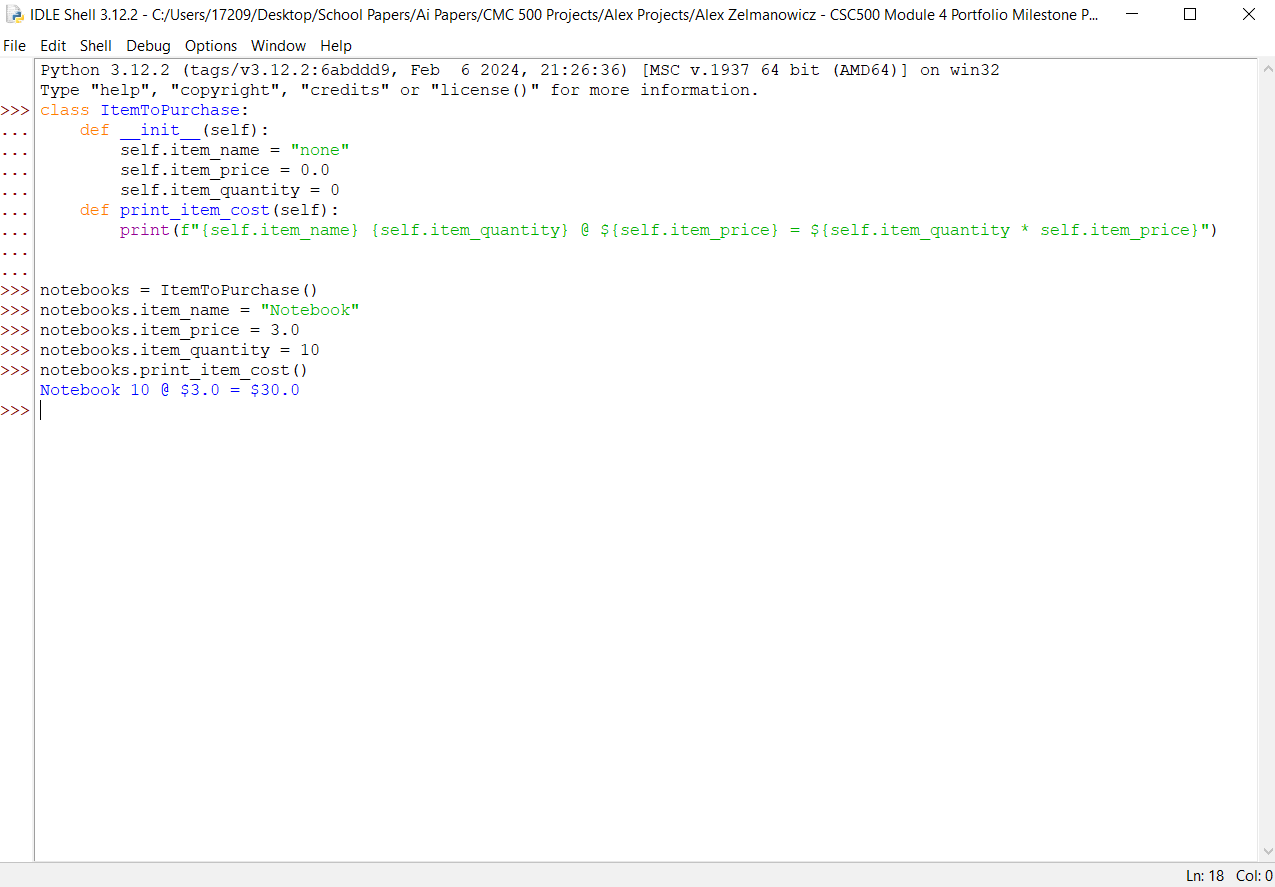
Colorado State University Global

Professor Farr

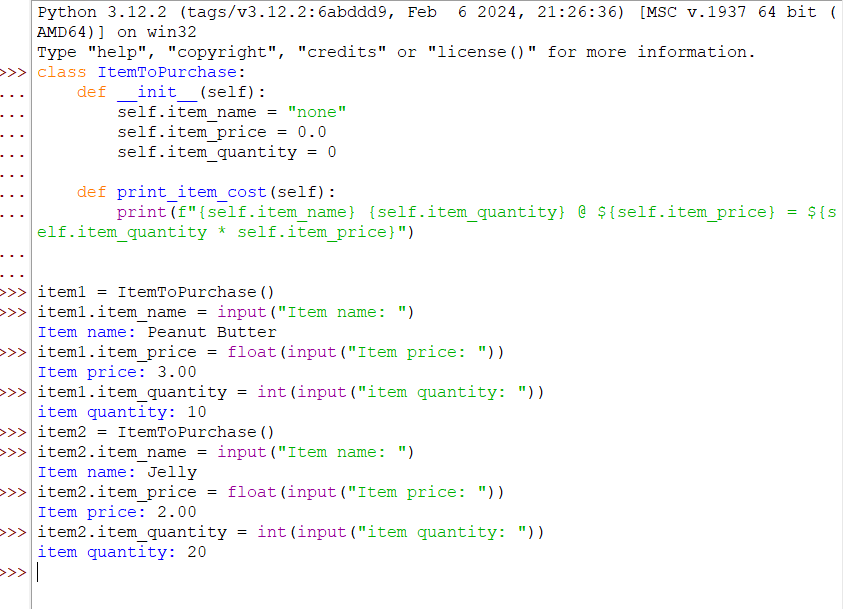
CSC 500

10 March 2024

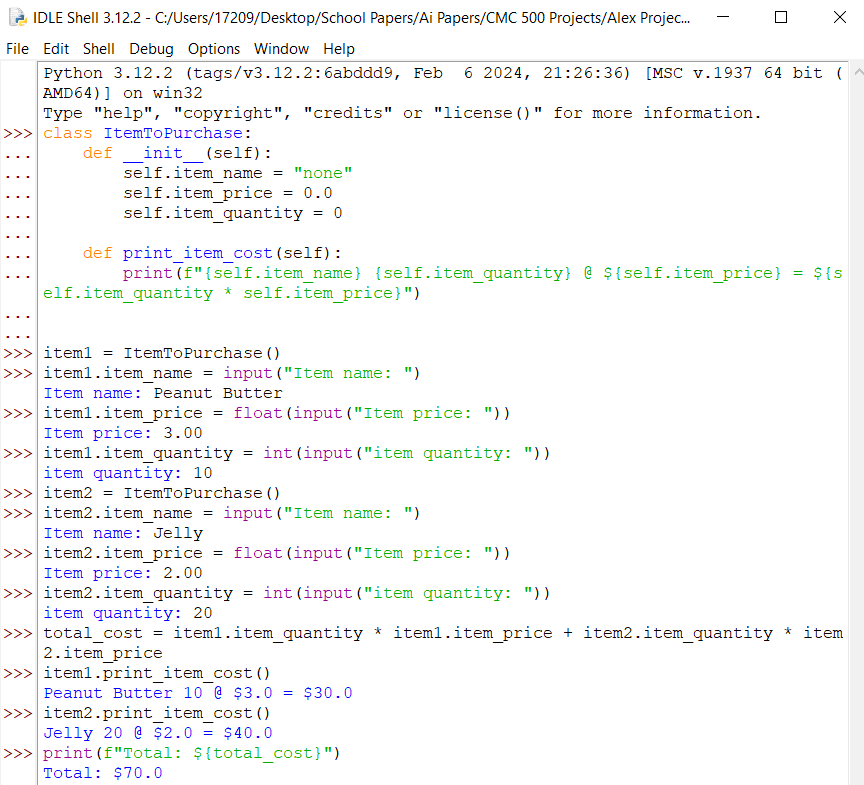
**Step 1:**

****

**Step 2:**

****

**Step 3:**

****

**Step 1 Source Code:**

>>> class ItemToPurchase:

... def \_\_init\_\_(self):

... self.item\_name = "none"

... self.item\_price = 0.0

... self.item\_quantity = 0

... def print\_item\_cost(self):

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

...

...

>>> notebooks = ItemToPurchase()

>>> notebooks.item\_name = "Notebook"

>>> notebooks.item\_price = 3.0

>>> notebooks.item\_quantity = 10

>>> notebooks.print\_item\_cost()

Notebook 10 @ $3.0 = $30.0 **Step 2 Source Code:**

>>> class ItemToPurchase:

... def \_\_init\_\_(self):

... self.item\_name = "none"

... self.item\_price = 0.0

... self.item\_quantity = 0

...

... def print\_item\_cost(self):

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

...

...

>>> item1 = ItemToPurchase()

>>> item1.item\_name = input("Item name: ")

Item name: Peanut Butter

>>> item1.item\_price = float(input("Item price: "))

Item price: 3.00

>>> item1.item\_quantity = int(input("item quantity: "))

item quantity: 10

>>> item2 = ItemToPurchase()

>>> item2.item\_name = input("Item name: ")

Item name: Jelly

>>> item2.item\_price = float(input("Item price: "))

Item price: 2.00

>>> item2.item\_quantity = int(input("item quantity: "))

item quantity: 20

**Step 3 Source Code:**

>>> class ItemToPurchase:

... def \_\_init\_\_(self):

... self.item\_name = "none"

... self.item\_price = 0.0

... self.item\_quantity = 0

...

... def print\_item\_cost(self):

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

...

...

>>> item1 = ItemToPurchase()

>>> item1.item\_name = input("Item name: ")

Item name: Peanut Butter

>>> item1.item\_price = float(input("Item price: "))

Item price: 3.00

>>> item1.item\_quantity = int(input("item quantity: "))

item quantity: 10

>>> item2 = ItemToPurchase()

>>> item2.item\_name = input("Item name: ")

Item name: Jelly

>>> item2.item\_price = float(input("Item price: "))

Item price: 2.00

>>> item2.item\_quantity = int(input("item quantity: "))

item quantity: 20

>>> total\_cost = item1.item\_quantity \* item1.item\_price + item2.item\_quantity \* item2.item\_price

>>> item1.print\_item\_cost()

Peanut Butter 10 @ $3.0 = $30.0

>>> item2.print\_item\_cost()

Jelly 20 @ $2.0 = $40.0

>>> print(f"Total: ${total\_cost}")

Total: $70.0

**Step 1, 2, and 3 Pseudocode:**1. Start.

2. Make a class called ItemToPurchase.

3. Input item\_name as "none", item\_price as 0.0, and item\_quantity as 0.

4. Make a method named print\_item\_cost.

5. Make an instance of ItemToPurchase called item1.

6. Ask the user for the name, price, and quantity of item1.

7. Make a second instance of ItemToPurchase called item2.

8. Ask the user for the name, price, and quantity of item2.

9. Calculate the cost of item 1 by its quantity multiplied by price.

10. Calculate the cost of item 1 by its quantity multiplied by price.

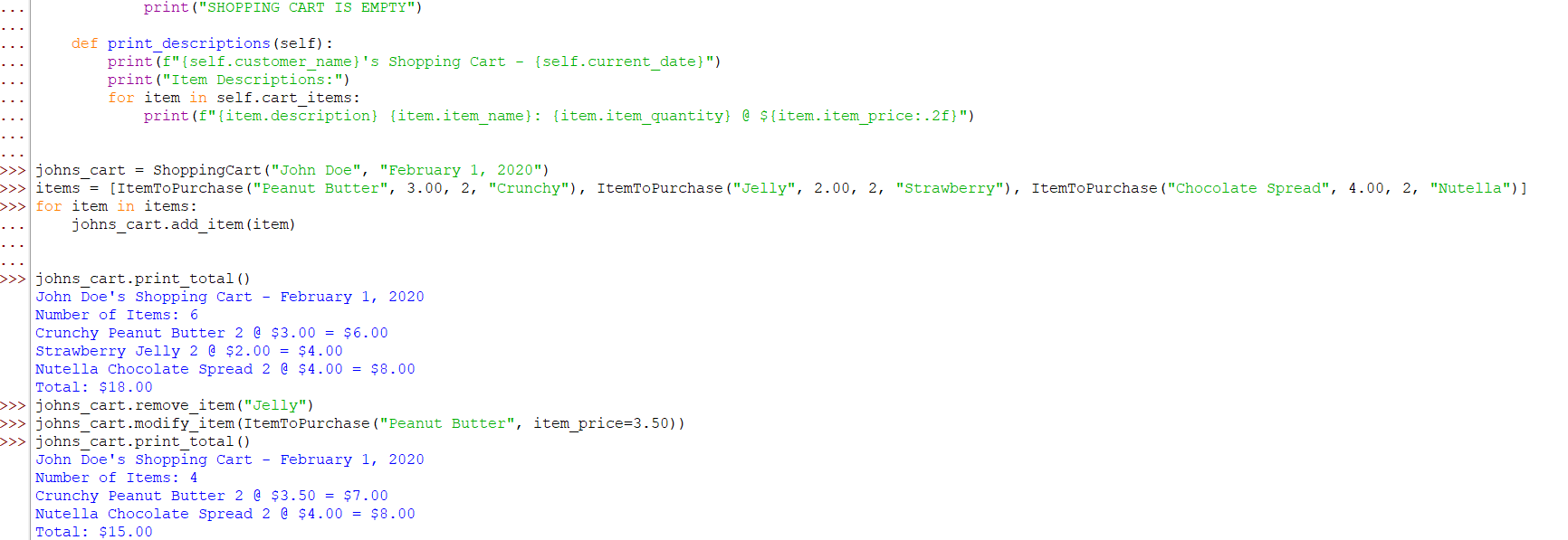
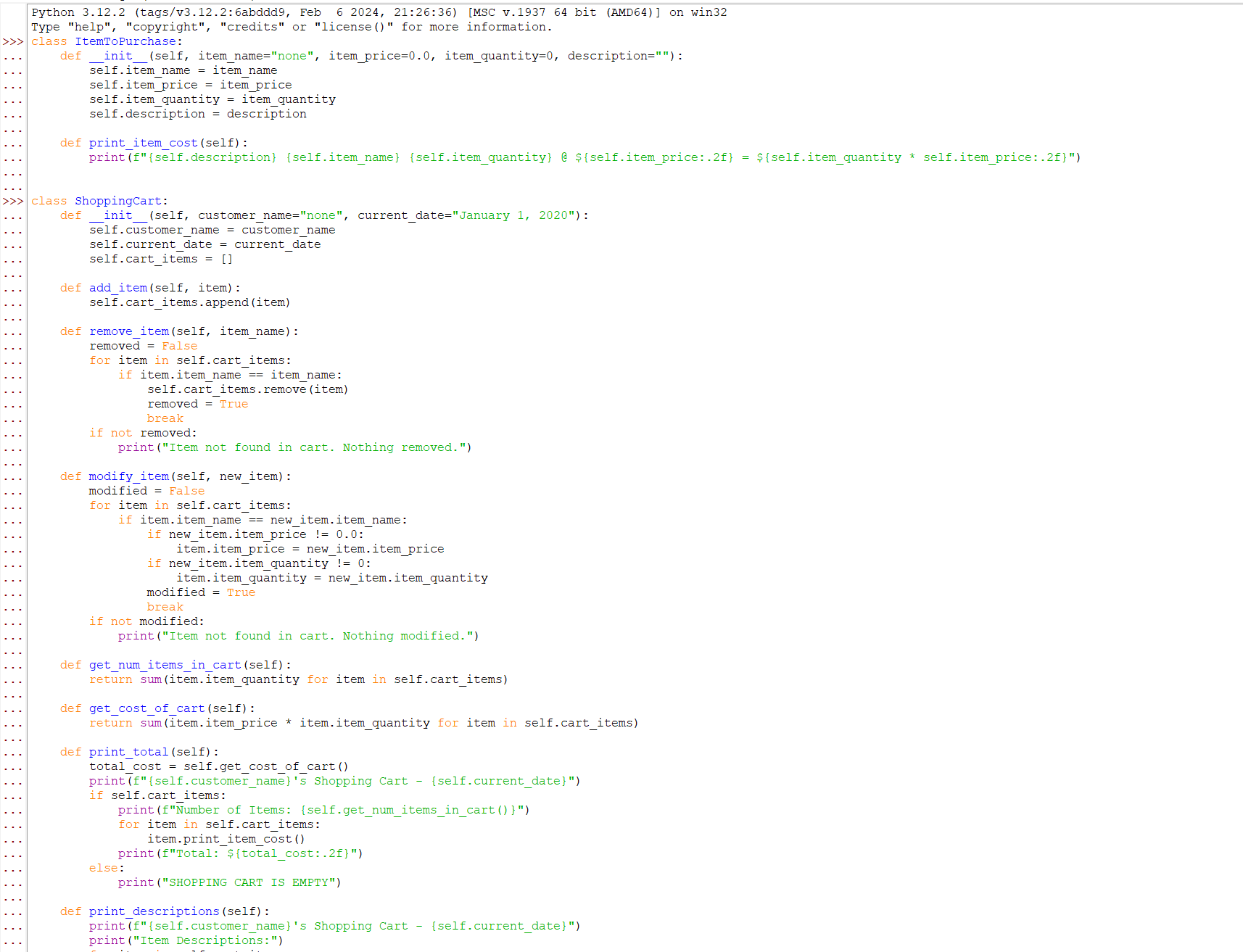
11. Print the details for item 1 and item 2.

12. find the total cost by adding both total prices of item 1 and item 2 by all its quantity.

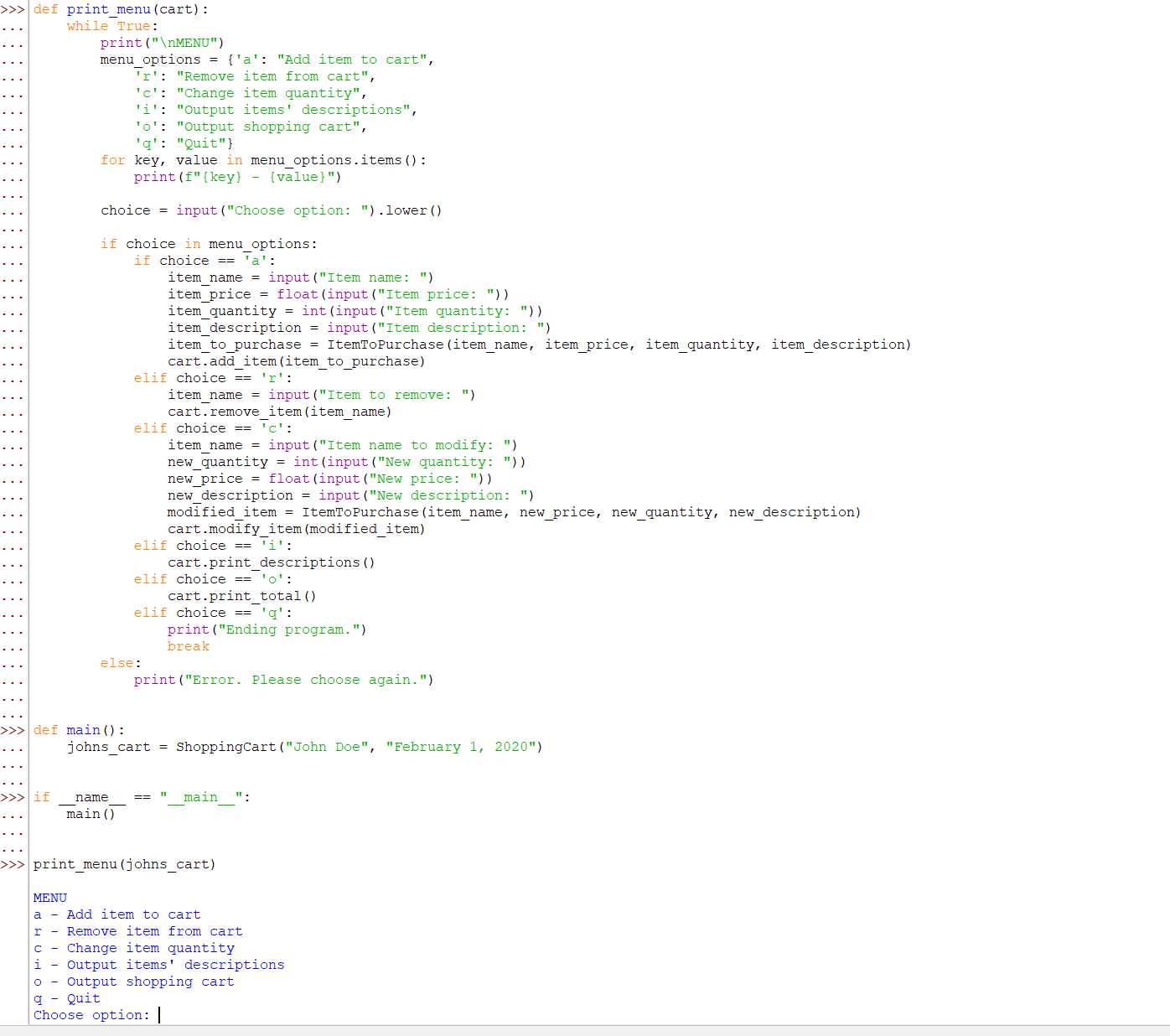
13. print the total cost of all items.

14. End.

**Step 4:**

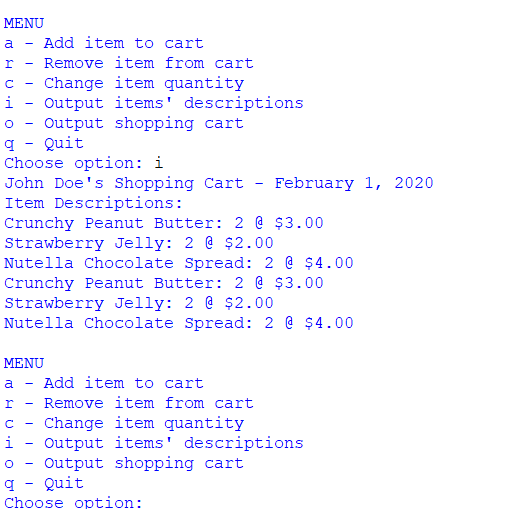
****

**Step 5:**

****

**Step 6:**

****



**Step 4 Source Code:**

>>> class ItemToPurchase:

... def \_\_init\_\_(self, item\_name="none", item\_price=0.0, item\_quantity=0, description=""):

... self.item\_name = item\_name

... self.item\_price = item\_price

... self.item\_quantity = item\_quantity

... self.description = description

...

... def print\_item\_cost(self):

... print(f"{self.description} {self.item\_name} {self.item\_quantity} @ ${self.item\_price:.2f} = ${self.item\_quantity \* self.item\_price:.2f}")

...

...

>>> 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):

... removed = False

... for item in self.cart\_items:

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

... self.cart\_items.remove(item)

... removed = True

... break

... if not removed:

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

...

... def modify\_item(self, new\_item):

... modified = False

... for item in self.cart\_items:

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

... if new\_item.item\_price != 0.0:

... item.item\_price = new\_item.item\_price

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

... item.item\_quantity = new\_item.item\_quantity

... modified = True

... break

... if not modified:

... 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):

... total\_cost = self.get\_cost\_of\_cart()

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

... if self.cart\_items:

... print(f"Number of Items: {self.get\_num\_items\_in\_cart()}")

... for item in self.cart\_items:

... item.print\_item\_cost()

... print(f"Total: ${total\_cost:.2f}")

... else:

... print("SHOPPING CART IS EMPTY")

...

... def print\_descriptions(self):

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

... print("Item Descriptions:")

... for item in self.cart\_items:

print(f"{item.description} {item.item\_name}: {item.item\_quantity} @ ${item.item\_price:.2f}")

johns\_cart = ShoppingCart("John Doe", "February 1, 2020")

items = [ItemToPurchase("Peanut Butter", 3.00, 2, "Crunchy"), ItemToPurchase("Jelly", 2.00, 2, "Strawberry"), ItemToPurchase("Chocolate Spread", 4.00, 2, "Nutella")]

for item in items:

johns\_cart.add\_item(item)

johns\_cart.print\_total()

John Doe's Shopping Cart - February 1, 2020

Number of Items: 6

Crunchy Peanut Butter 2 @ $3.00 = $6.00

Strawberry Jelly 2 @ $2.00 = $4.00

Nutella Chocolate Spread 2 @ $4.00 = $8.00

Total: $18.00

johns\_cart.remove\_item("Jelly")

johns\_cart.modify\_item(ItemToPurchase("Peanut Butter", item\_price=3.50))

johns\_cart.print\_total()

John Doe's Shopping Cart - February 1, 2020

Number of Items: 4

Crunchy Peanut Butter 2 @ $3.50 = $7.00

Nutella Chocolate Spread 2 @ $4.00 = $8.00

Total: $15.00 **Step 5 Source Code:**

class ItemToPurchase:

def \_\_init\_\_(self, item\_name="none", item\_price=0.0, item\_quantity=0, description=""):

self.item\_name = item\_name

self.item\_price = item\_price

self.item\_quantity = item\_quantity

self.description = description

def print\_item\_cost(self):

print(f"{self.description} {self.item\_name} {self.item\_quantity} @ ${self.item\_price:.2f} = ${self.item\_quantity \* self.item\_price:.2f}")

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):

removed = False

for item in self.cart\_items:

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

self.cart\_items.remove(item)

removed = True

break

if not removed:

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

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

modified = False

for item in self.cart\_items:

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

if new\_item.item\_price != 0.0:

item.item\_price = new\_item.item\_price

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

item.item\_quantity = new\_item.item\_quantity

modified = True

break

if not modified:

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):

total\_cost = self.get\_cost\_of\_cart()

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

if self.cart\_items:

print(f"Number of Items: {self.get\_num\_items\_in\_cart()}")

for item in self.cart\_items:

item.print\_item\_cost()

print(f"Total: ${total\_cost:.2f}")

else:

print("SHOPPING CART IS EMPTY")

def print\_descriptions(self):

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

print("Item Descriptions:")

for item in self.cart\_items:

print(f"{item.description} {item.item\_name}: {item.item\_quantity} @ ${item.item\_price:.2f}")

johns\_cart = ShoppingCart("John Doe", "February 1, 2020")

items = [ItemToPurchase("Peanut Butter", 3.00, 2, "Crunchy"), ItemToPurchase("Jelly", 2.00, 2, "Strawberry"), ItemToPurchase("Chocolate Spread", 4.00, 2, "Nutella")]

for item in items:

johns\_cart.add\_item(item)

def print\_menu(cart):

while True:

print("\nMENU")

menu\_options = {'a': "Add item to cart",

'r': "Remove item from cart",

'c': "Change item quantity",

'i': "Output items' descriptions",

'o': "Output shopping cart",

'q': "Quit"}

for key, value in menu\_options.items():

print(f"{key} - {value}")

choice = input("Choose option: ").lower()

if choice in menu\_options:

if choice == 'a':

item\_name = input("Item name: ")

item\_price = float(input("Item price: "))

item\_quantity = int(input("Item quantity: "))

item\_description = input("Item description: ")

item\_to\_purchase = ItemToPurchase(item\_name, item\_price, item\_quantity, item\_description)

cart.add\_item(item\_to\_purchase)

... elif choice == 'r':

... item\_name = input("Item to remove: ")

... cart.remove\_item(item\_name)

... elif choice == 'c':

... item\_name = input("Item name to modify: ")

... new\_quantity = int(input("New quantity: "))

... new\_price = float(input("New price: "))

... new\_description = input("New description: ")

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

... cart.modify\_item(modified\_item)

... elif choice == 'i':

... cart.print\_descriptions()

... elif choice == 'o':

... cart.print\_total()

... elif choice == 'q':

... print("Ending program.")

... break

... else:

... print("Error. Please choose again.")

...

...

>>> def main():

... johns\_cart = ShoppingCart("John Doe", "February 1, 2020")

...

...

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

... main()

...

...

>>> print\_menu(johns\_cart)

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

**Step 6 Source Code:**

print\_menu(johns\_cart)

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 option: a

Item name: Peanut Butter

Item price: 3.00

Item quantity: 2

Item description: Crunchy

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 option: a

Item name: Jelly

Item price: 2.00

Item quantity: 2

Item description: Strawberry

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 option: a

Item name: Chocolate Spread

Item price: 4.00

Item quantity: 2

Item description: Nutella

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 option: o

John Doe's Shopping Cart - February 1, 2020

Number of Items: 12

Crunchy Peanut Butter 2 @ $3.00 = $6.00

Strawberry Jelly 2 @ $2.00 = $4.00

Nutella Chocolate Spread 2 @ $4.00 = $8.00

Crunchy Peanut Butter 2 @ $3.00 = $6.00

Strawberry Jelly 2 @ $2.00 = $4.00

Nutella Chocolate Spread 2 @ $4.00 = $8.00

Total: $36.00

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 option: i

John Doe's Shopping Cart - February 1, 2020

Item Descriptions:

Crunchy Peanut Butter: 2 @ $3.00

Strawberry Jelly: 2 @ $2.00

Nutella Chocolate Spread: 2 @ $4.00

Crunchy Peanut Butter: 2 @ $3.00

Strawberry Jelly: 2 @ $2.00

Nutella Chocolate Spread: 2 @ $4.00

MENU

a - Add item to cart

r - Remove item from cart

c - Change item quantity

i - Output items' descriptions

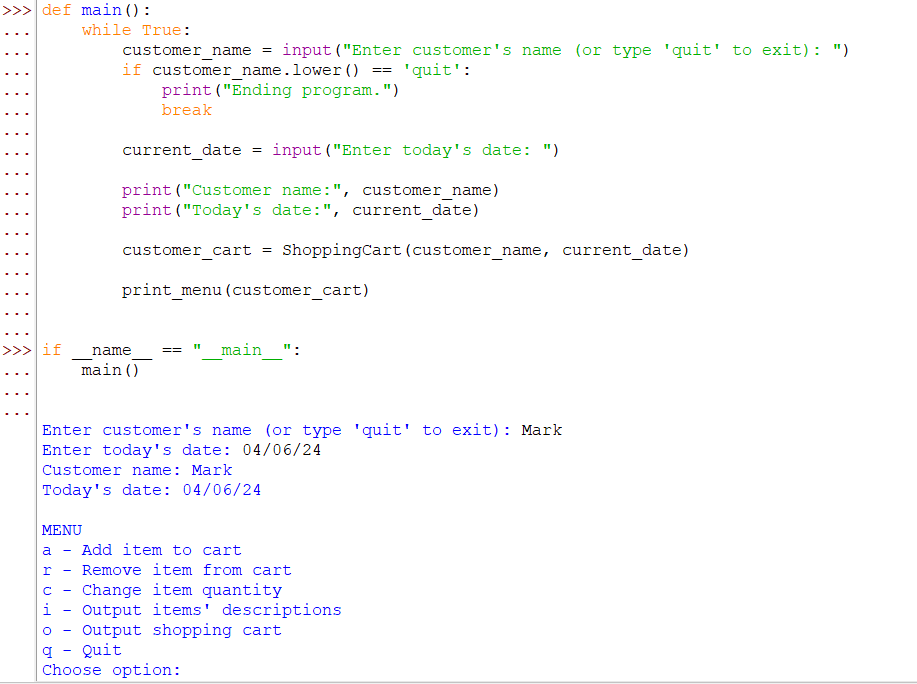
o - Output shopping cart

q - Quit

Choose option: q

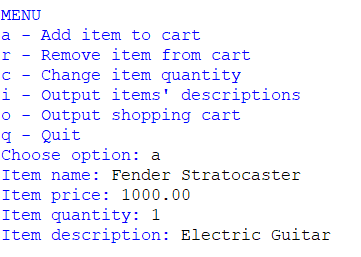
Ending program.

**Step 7:**

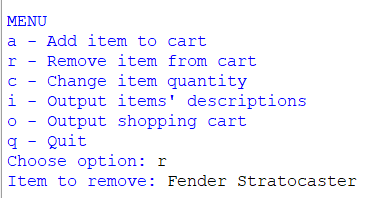


**Git**

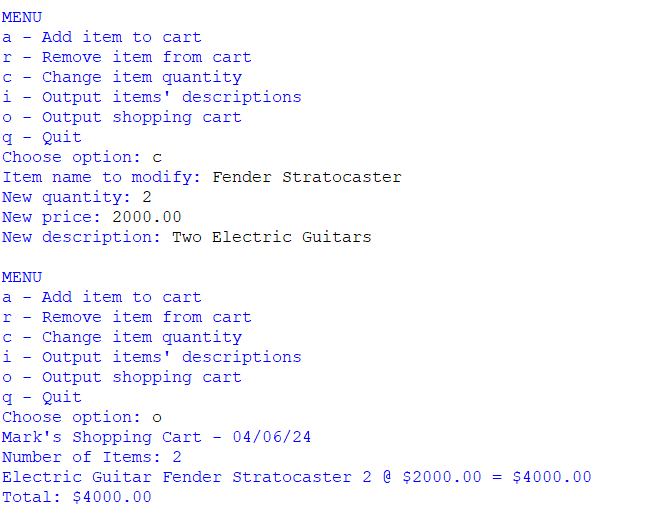
**Step 8:**



**Step 9:**



**Step 10:**



**Source Code:**

Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32

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

class ItemToPurchase:

def \_\_init\_\_(self, item\_name="none", item\_price=0.0, item\_quantity=0, description=""):

self.item\_name = item\_name

self.item\_price = item\_price

self.item\_quantity = item\_quantity

self.description = description

def print\_item\_cost(self):

print(f"{self.description} {self.item\_name} {self.item\_quantity} @ ${self.item\_price:.2f} = ${self.item\_quantity \* self.item\_price:.2f}")

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):

removed = False

for item in self.cart\_items:

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

self.cart\_items.remove(item)

removed = True

break

if not removed:

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

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

modified = False

for item in self.cart\_items:

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

if new\_item.item\_price != 0.0:

item.item\_price = new\_item.item\_price

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

item.item\_quantity = new\_item.item\_quantity

modified = True

break

if not modified:

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):

total\_cost = self.get\_cost\_of\_cart()

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

if self.cart\_items:

print(f"Number of Items: {self.get\_num\_items\_in\_cart()}")

for item in self.cart\_items:

item.print\_item\_cost()

print(f"Total: ${total\_cost:.2f}")

else:

print("SHOPPING CART IS EMPTY")

def print\_descriptions(self):

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

print("Item Descriptions:")

for item in self.cart\_items:

print(f"{item.description} {item.item\_name}: {item.item\_quantity} @ ${item.item\_price:.2f}")

def print\_menu(cart):

while True:

print("\nMENU")

menu\_options = {'a': "Add item to cart",

'r': "Remove item from cart",

'c': "Change item quantity",

'i': "Output items' descriptions",

'o': "Output shopping cart",

'q': "Quit"}

for key, value in menu\_options.items():

print(f"{key} - {value}")

choice = input("Choose option: ").lower()

if choice in menu\_options:

if choice == 'a':

item\_name = input("Item name: ")

item\_price = float(input("Item price: "))

item\_quantity = int(input("Item quantity: "))

item\_description = input("Item description: ")

item\_to\_purchase = ItemToPurchase(item\_name, item\_price, item\_quantity, item\_description)

cart.add\_item(item\_to\_purchase)

elif choice == 'r':

item\_name = input("Item to remove: ")

cart.remove\_item(item\_name)

elif choice == 'c':

item\_name = input("Item name to modify: ")

new\_quantity = int(input("New quantity: "))

new\_price = float(input("New price: "))

new\_description = input("New description: ")

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

cart.modify\_item(modified\_item)

elif choice == 'i':

cart.print\_descriptions()

elif choice == 'o':

cart.print\_total()

elif choice == 'q':

print("Ending program.")

break

else:

print("Error. Please choose again.")

def main():

while True:

customer\_name = input("Enter customer's name (or type 'quit' to exit): ")

if customer\_name.lower() == 'quit':

print("Ending program.")

break

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

print("Customer name:", customer\_name)

print("Today's date:", current\_date)

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

print\_menu(customer\_cart)

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

main()

Enter customer's name (or type 'quit' to exit): Mark

Enter today's date: 04/06/24

Customer name: Mark

Today's date: 04/06/24

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 option: a

Item name: Fender Stratocaster

Item price: 1000.00

Item quantity: 1

Item description: Electric Guitar

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 option: r

Item to remove: Fender Stratocaster

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 option: a

Item name: Fender Stratocaster

Item price: 1000.00

Item quantity: 1

Item description: Electric Guitar

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 option: c

Item name to modify: Fender Stratocaster

New quantity: 2

New price: 2000.00

New description: Two Electric Guitars

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 option: o

Mark's Shopping Cart - 04/06/24

Number of Items: 2

Electric Guitar Fender Stratocaster 2 @ $2000.00 = $4000.00

Total: $4000.00

**Pseudocode 7, 8, 9, and 10:**

**1.** Start

**2.** Create classes called ItemToPurchase and ShoppingCart.

**3.** In ItemToPurchase indicate the name price quantity and description of item.

**4.** In ShoppingCart indicate the name, current date and an empty list in the cart.

**5.** Define a print\_menu function that can add remove and modify items (such as quantity) and gives descriptions of the shopping cart.

**6.** Create a shopping cart object for the customer.

**7.** add the object to cart.

**8.** remove object from cart.

**9.** modify the quantity and price.

**10.** Output shopping cart.

**11.** Exit the program.

**12.** End.

**Repository**: <https://github.com/AlexCSUGlobal/CSC500>