Module 6 Portfolio Project

Alex Zelmanowicz

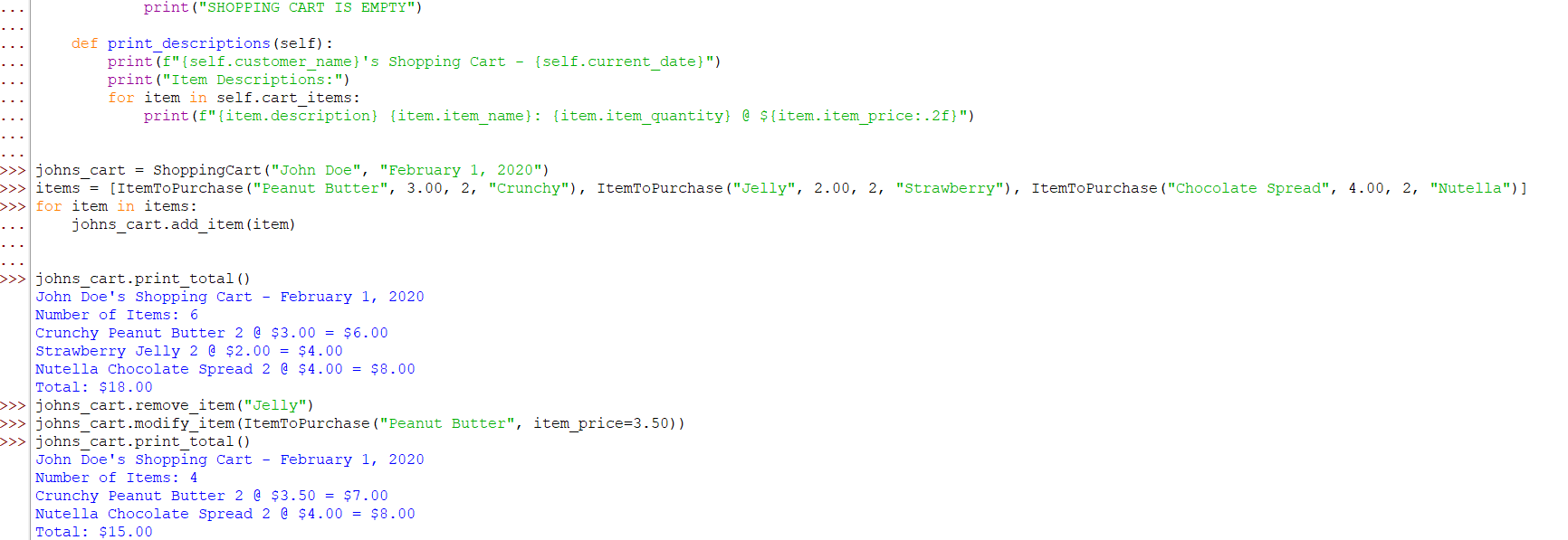
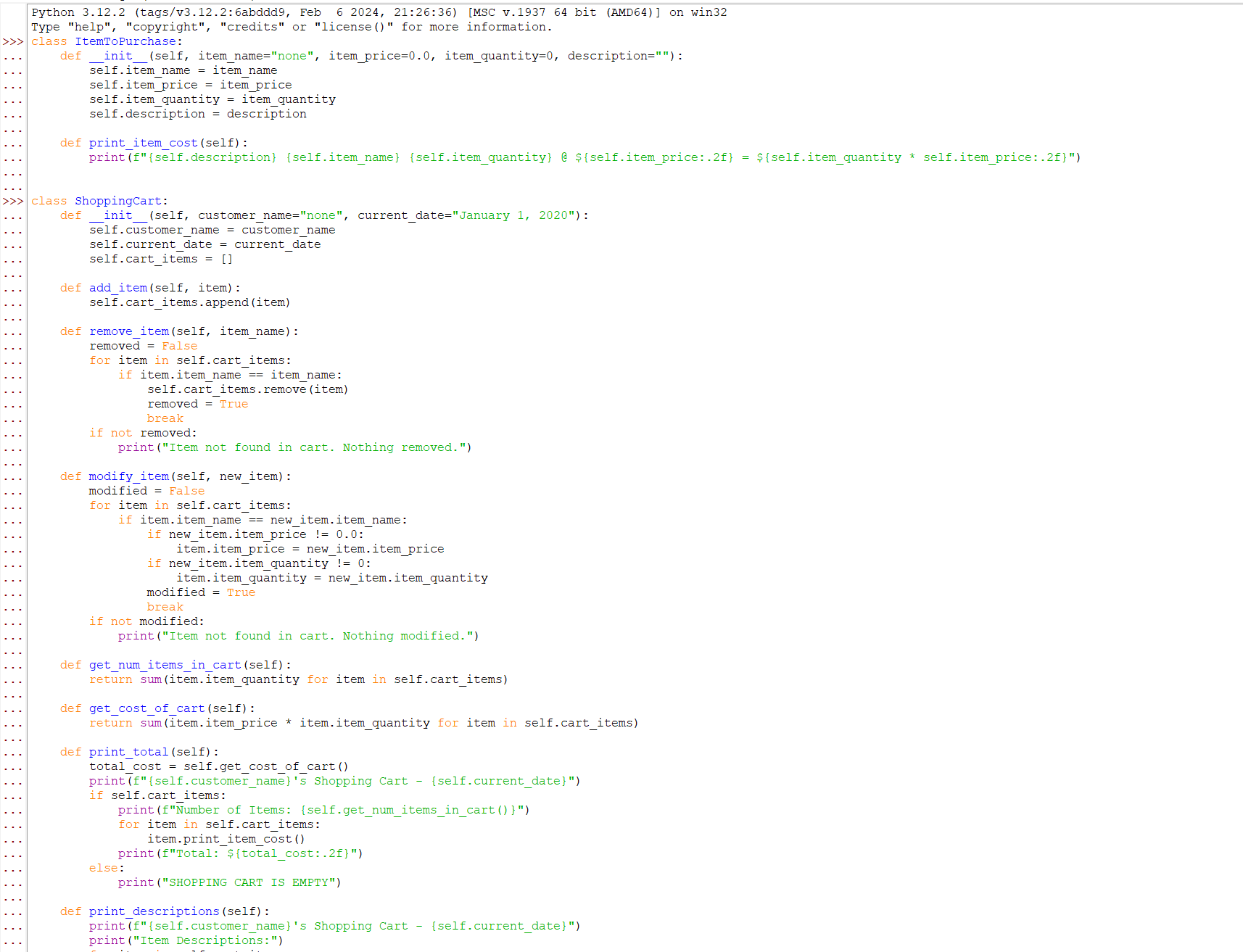
Colorado State University Global

Professor Farr

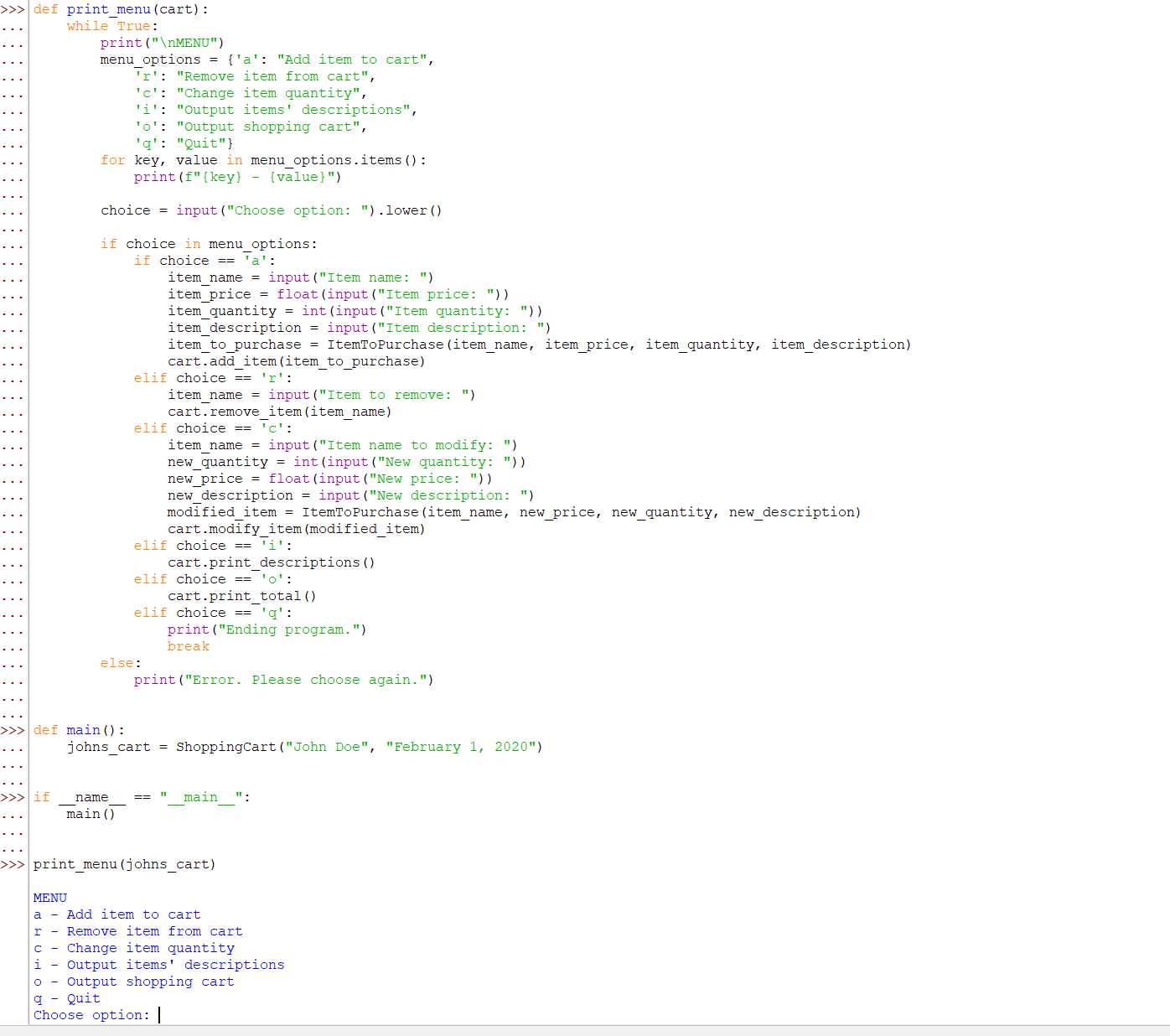
CSC 500

25 March 2024

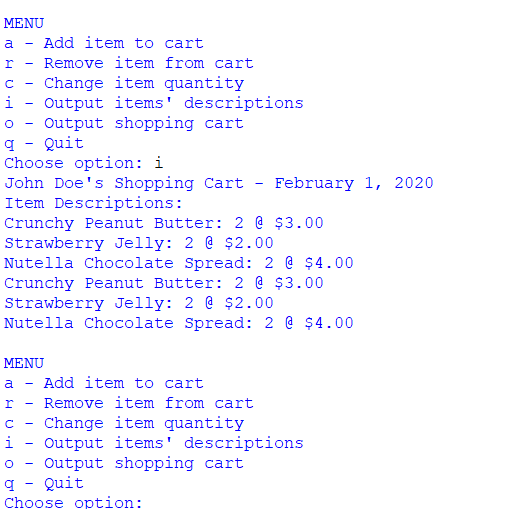
**Step 4:**

****

**Step 2:**

****

**Step 3:**



**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.

**Git Repository**: https://github.com/AlexCSUGlobal/CSC500