**Pseudocode:**

CLASS: ItemToPurchase

METHOD \_\_init\_\_(name="none", price=0, quantity=0, description="none")

SET item\_name TO name

SET item\_price TO price

SET item\_quantity TO quantity

SET item\_description TO description

METHOD print\_item\_cost()

total = item\_quantity \* item\_price

PRINT item\_name, item\_quantity, item\_price, and total

METHOD print\_item\_description()

PRINT item\_name and item\_description

CLASS: ShoppingCart

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

SET customer\_name

SET current\_date

INITIALIZE cart\_items AS empty list

METHOD add\_item(item)

APPEND item TO cart\_items

METHOD remove\_item(item\_name)

IF item\_name EXISTS IN cart\_items

REMOVE item

ELSE

PRINT "Item not found in cart. Nothing removed."

METHOD modify\_item(item)

IF item\_name EXISTS IN cart\_items

IF item.description/price/quantity IS NOT default

UPDATE respective fields

ELSE

PRINT "Item not found in cart. Nothing modified."

METHOD get\_num\_items\_in\_cart()

RETURN SUM of item\_quantity FOR each item

METHOD get\_cost\_of\_cart()

RETURN SUM of (item\_quantity \* item\_price) FOR each item

METHOD print\_total()

PRINT customer\_name, current\_date

PRINT each item's cost

PRINT total cost or "SHOPPING CART IS EMPTY"

METHOD print\_descriptions()

PRINT customer\_name, current\_date

PRINT each item's description

FUNCTION print\_menu(cart)

WHILE user\_input IS NOT 'q'

DISPLAY menu options

GET user choice

CASE 'a': PROMPT and ADD item

CASE 'r': PROMPT and REMOVE item

CASE 'c': PROMPT and MODIFY item quantity

CASE 'i': PRINT item descriptions

CASE 'o': PRINT shopping cart

CASE 'q': EXIT

DEFAULT: CONTINUE loop

FUNCTION main()

PROMPT for customer\_name and current\_date

PRINT customer\_name and current\_date

CREATE ShoppingCart object

CALL print\_menu(cart)

**Code:**

class ItemToPurchase:

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

        self.item\_name = item\_name

        self.item\_price = float(item\_price)

        self.item\_quantity = int(item\_quantity)

        self.item\_description = item\_description

    def print\_item\_cost(self):

        total = self.item\_quantity \* self.item\_price

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

    def print\_item\_description(self):

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

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):

        found = False

        for i, item in enumerate(self.cart\_items):

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

                del self.cart\_items[i]

                found = True

                break

        if not found:

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

    def modify\_item(self, item):

        found = False

        for cart\_item in self.cart\_items:

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

                found = True

                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

                break

        if not found:

            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\_quantity \* item.item\_price for item in self.cart\_items)

    def print\_total(self):

        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 not self.cart\_items:

            print("SHOPPING CART IS EMPTY")

        else:

            for item in self.cart\_items:

                item.print\_item\_cost()

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

    def print\_descriptions(self):

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

        print("Item Descriptions")

        for item in self.cart\_items:

            item.print\_item\_description()

def print\_menu(cart):

    menu = (

        "\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"

    )

    choice = ""

    while choice != "q":

        print(menu)

        choice = input("Choose an option:\n").strip().lower()

        if choice == "a":

            print("ADD ITEM TO CART")

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

            desc = input("Enter the item description:\n")

            price = float(input("Enter the item price:\n"))

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

            cart.add\_item(ItemToPurchase(name, price, quantity, desc))

        elif choice == "r":

            print("REMOVE ITEM FROM CART")

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

            cart.remove\_item(name)

        elif choice == "c":

            print("CHANGE ITEM QUANTITY")

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

            quantity = int(input("Enter the new quantity:\n"))

            cart.modify\_item(ItemToPurchase(item\_name=name, item\_quantity=quantity))

        elif choice == "i":

            print("OUTPUT ITEMS' DESCRIPTIONS")

            cart.print\_descriptions()

        elif choice == "o":

            print("OUTPUT SHOPPING CART")

            cart.print\_total()

        elif choice == "q":

            break

        else:

            continue

def main():

    print("Enter customer's name:")

    customer\_name = input()

    print("Enter today's date:")

    current\_date = input()

    print()

    print(f"Customer name: {customer\_name}")

    print(f"Today's date: {current\_date}")

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

    print\_menu(cart)

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

    main()

**Screenshots:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**GitHub:**

<https://github.com/mnem0nic7/CSC500/tree/main/Portfolio%20Milestone%208>