module6

December 16, 2025

Assignment “‘Online Shopping Cart Step 1: Build the ItemToPurchase class with the following specifications:

Attributes item_name (string) item_price (float) item_quantity (int) Default constructor Initializes item’s name = “none”, item’s price = 0, item’s quantity = 0 Method print_item_cost() Example of print_item_cost() output: Bottled Water 10 @ $1 = $10

Step 2: In the main section of your code, prompt the user for two items and create two objects of the ItemToPurchase class.

Example:

Item 1

Enter the item name:

Chocolate Chips

Enter the item price:

3

Enter the item quantity:

1

Item 2

Enter the item name:

Bottled Water

Enter the item price:

1

Enter the item quantity:

10

Step 3: Add the costs of the two items together and output the total cost.

Example:

TOTAL COST

Chocolate Chips 1 @ $3 = $3

Bottled Water 10 @ $1 = $10

Total: $13

“‘

Git Repository https://github.com/izzy64/CSC500/blob/main/Module4

CODE

[11]:

```
# Part 1

class ItemToPurchase:
    def __init__(self, item_name="none", item_price=0, item_quantity=0,␣
 ↪item_description="none"):
        """Default constructor."""
        self.item_name = str(item_name)
        self.item_price = float(item_price)
        self.item_quantity = int(item_quantity)
        self.item_description = str(item_description)

    def print_item_cost(self):
        """Print Item Cost."""
        return f"{self.item_name} {self.item_quantity} @ ${self.item_price:.2f}␣
 ↪= ${(self.item_price*self.item_quantity):.2f}"
```

[16]:

```
# Part 2

item_count = 2
items = []

for i in range(1,item_count+1):
    print(f"Item {i}")
    print(f"Enter the item name:")
    item_name = str(input())
    print(item_name)
    print(f"Enter the item price:")
    item_price = round(float(input()), 2)
    print(item_price)
    print(f"Enter the item quantity:")
    item_quantity = int(input())
    print(item_quantity)
    items.append(ItemToPurchase(item_name, item_price, item_quantity))
```

```
Item 1
Enter the item name:
Chocolate Chips
Enter the item price:
3.24
Enter the item quantity:
```

```
4
Item 2
Enter the item name:
Bottled Water
Enter the item price:
1.96
Enter the item quantity:
6
```

[18]:

```
# Step 3

print("TOTAL COST\n")

total = 0
for item in items:
    print(item.print_item_cost(), "\n")
    total += item.item_price*item.item_quantity

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

```
TOTAL COST

Chocolate Chips 4 @ $3.24 = $12.96

Bottled Water 6 @ $1.96 = $11.76

Total: $24.72
```

Step 4: Build the ShoppingCart class with the following data attributes and related methods. Note: Some can be method stubs (empty methods) initially, to be completed in later steps

Parameterized constructor, which takes the customer name and date as parameters Attributes
customer_name (string) - Initialized in default constructor to “none”
current_date (string) - Initialized in default constructor to “January 1, 2020” cart_items (list)
Methods
add_item()
Adds an item to cart_items list. Has parameter ItemToPurchase. Does not return anything.
remove_item()
Removes item from cart_items list. Has a string (an item’s name) parameter. Does not return anything.
If item name cannot be found, output this message: 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.
If item can be found (by name) in cart, check if parameter has default values for description, price, and quantity. If not, modify item in cart.
If item cannot be found (by name) in cart, output this message: Item not found in cart. Nothing modified.
get_num_items_in_cart()

Returns quantity of all items in cart. Has no parameters.
get_cost_of_cart()
Determines and returns the total cost of items in cart. Has no parameters.
print_total()
Outputs total of objects in cart.
If cart is empty, output this message: SHOPPING CART IS EMPTY
print_descriptions()
Outputs each item's description.
Example of print_total() output:
John Doe's Shopping Cart - February 1, 2020 Number of Items: 8
Nike Romaleos 2 @ $189 = $378
Chocolate Chips 5 @ $3 = $15
Powerbeats 2 Headphones 1 @ $128 = $128 Total: $521

Example of print_descriptions() output: John Doe's Shopping Cart - February 1, 2020 Item Descriptions
Nike Romaleos: Volt color, Weightlifting shoes
Chocolate Chips: Semi-sweet Powerbeats 2 Headphones: Bluetooth headphones

[12]:

```python
class ShoppingCart:
    def __init__(self, customer_name="none", current_date="January 1, 2026"):
        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):
        for item in self.cart_items:
            if item.item_name == item_name:
                self.cart_items.remove(item)
                return
        print("Item not found in cart. Nothing removed.")

    def modify_item(self, item_to_modify):
        for item in self.cart_items:
            if item.item_name == item_to_modify.item_name:
                # Only modify if the new values are not default values
                if item_to_modify.item_price != 0:
                    item.item_price = item_to_modify.item_price
                if item_to_modify.item_quantity != 0:
                    item.item_quantity = item_to_modify.item_quantity
                return
        print("Item not found in cart. Nothing modified.")

    def get_num_items_in_cart(self):
        total_quantity = 0
        for item in self.cart_items:
            total_quantity += item.item_quantity
        return total_quantity

    def get_cost_of_cart(self):
        total_cost = 0
        for item in self.cart_items:
            total_cost += item.item_price * item.item_quantity
        return total_cost

    def print_total(self):
        print(f"{self.customer_name}'s Shopping Cart - {self.current_date}")
        print(f"Number of Items: {self.get_num_items_in_cart()}\n")

        if len(self.cart_items) == 0:
            print("SHOPPING CART IS EMPTY")
        else:
            for item in self.cart_items:
                print(item.print_item_cost())

        print(f"\nTotal: ${self.get_cost_of_cart():.2f}")

    def print_descriptions(self):
        print(f"{self.customer_name}'s Shopping Cart - {self.current_date}\n")
        print("Item Descriptions")
        for item in self.cart_items:
            print(f"{item.item_name}: {item.item_description}")
```

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. Each option is represented by a single character. Build and output the menu within the function.

If an invalid character is entered, continue to prompt for a valid choice. Hint: Implement Quit before implementing other options. Call print_menu() in the main() function. Continue to execute the menu until the user enters q to Quit.

Example:
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:

Step 6: Implement Output shopping cart menu option. Implement Output item’s description menu option.

Example of shopping cart menu option:

OUTPUT SHOPPING CART
John Doe's Shopping Cart - February 1, 2020 Number of Items: 8
Nike Romaleos 2 @ $189 = $378
Chocolate Chips 5 @ $3 = $15
Powerbeats 2 Headphones 1 @ $128 = $128 Total: $521

Example of item description menu option.
OUTPUT ITEMS' DESCRIPTIONS
John Doe's Shopping Cart - February 1, 2020 Item Descriptions
Nike Romaleos: Volt color, Weightlifting shoes
Chocolate Chips: Semi-sweet Powerbeats 2 Headphones: Bluetooth headphones

[17]:

```python
def print_menu(cart: ShoppingCart):
    select = ""
    choices = ["a", "r", "c", "i", "o", "q"]
    menu_txt = """
    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:
    """
    while select != "q":
        print(menu_txt)
        select = input()

        if select == "a":
            print("ADD ITEM TO CART")
            print("Enter the item name:")
            item_name = input()
            print("Enter the item description:")
            item_description = input()
            print("Enter the item price:")
            item_price = float(input())
            print("Enter the item quantity:")
            item_quantity = int(input())

            new_item = ItemToPurchase(item_name, item_price, item_quantity, item_description)
            cart.add_item(new_item)
            print("Item added!")

        elif select == "r":
            print("REMOVE ITEM FROM CART")
            print("Enter name of item to remove:")
            item_name = input()
            cart.remove_item(item_name)
            print("Item removed!")

        elif select == "c":
            print("CHANGE ITEM QUANTITY")
            print("Enter the item name:")
            item_name = input()
            print("Enter the new quantity:")
            new_quantity = int(input())

            item_to_modify = ItemToPurchase(item_name, 0, new_quantity)
            cart.modify_item(item_to_modify)
            print("Quantity changed!")

        elif select == "i":
            print("OUTPUT ITEMS' DESCRIPTIONS")
            cart.print_descriptions()

        elif select == "o":
            print("OUTPUT SHOPPING CART")
            cart.print_total()

        elif select == "q":
            print("THANKS FOR SHOPPING!")
            pass  # Exit the loop

        elif select not in choices:
            print("Invalid option. Please try again.")
        print("\n")
```

[19]:

```
cart = ShoppingCart("John Doe", "February 1, 2020")
print_menu(cart) # creating the 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 an option:

Invalid option. Please try again.
```

```
    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:

ADD ITEM TO CART
Enter the item name:
Enter the item description:
Enter the item price:
Enter the item quantity:
Item added!



    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:

ADD ITEM TO CART
Enter the item name:
Enter the item description:
Enter the item price:
Enter the item quantity:
Item added!



    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:

ADD ITEM TO CART
Enter the item name:
Enter the item description:
```

```
Enter the item price:
Enter the item quantity:
Item added!



    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:

ADD ITEM TO CART
Enter the item name:
Enter the item description:
Enter the item price:
Enter the item quantity:
Item added!



    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:

REMOVE ITEM FROM CART
Enter name of item to remove:
Item removed!



    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:
```

```
THANKS FOR SHOPPING!
```

[20]:

```
print_menu(cart)  # printing the 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 an option:

OUTPUT SHOPPING CART
John Doe's Shopping Cart - February 1, 2020
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:

OUTPUT ITEMS' DESCRIPTIONS
John Doe's Shopping Cart - February 1, 2020

Item Descriptions
Nike Romaleos: Volt color, Weightlifting shoes
Chocolate Chips: Semi-sweet
Powerbeats 2 Headphones: Bluetooth headphones



    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:

THANKS FOR SHOPPING!
```