Module 06: Portfolio Milestone

Saravenus Khon

Colorado State University Global

CSC500-1: Principles of Programming

Dr. Isaac K. Gang

May 08, 2025

Module 06: Portfolio Milestone

Git Repository

https://github.com/saraKhon/25SA-CSC500-1_Moduel-6_PortfolioMileStone.git

Part I: Source Code:

```
class ItemToPurchase:
    # Constructor method to initialize its default properties
    def __init__(self, name = "none", description = "none", price = 0, quantity =0):
        self.item_name = name              #Sting
        self.item_price = price            #float
        self.item_quantity = quantity      #int
        self.item_description = description   #string


    # Method NOT function: multiply and return the total of the item (price * quantity)
    def total_cost(self):
        return self.item_price * self.item_quantity

# Class: ShoppingCart
#Parameterized constructor, which takes the customer name and date as parameters
class ShoppingCart:
    def __init__(self, customer_name = "none", current_date = "January 1,2020"):
        self.customer_name = customer_name     #string, like a name tag on the shopping cart
        self.current_date = current_date       # string, the current date on the shopping cart
        self.cart_items = [ ]                  # empty list, empty shopping cart

# methods are below
# take the item and add it to the shopping cart. Doesn't return anything
    def add_item(self, item):
        self.cart_items.append(item)

# Removes item from the cart. Has a string (an item's name) parameter. Does not return anything.
# If the item name cannot be found, output this message: Item not found in cart. Nothing removed.
def remove_item(self, item_name):
        for index, item in enumerate(self.cart_items):  # for the position of the object, “item” on the cart
         if item.item_name.strip() ==  item_name. Strip():
              del self.cart_items[index]
              return
         print("No item founding cart. Nothing was removed")
```

Module 06: Portfolio Milestone

```
#getter method
def get_cart_items(self):
            return self.cart_items

def modify_item(self, item_to_modify):
    # for loop to index through each item in the cart
    for item in self.cart_items:
    # checks to see if the name of the current item matches the item to change
        if item.item_name.lower() == item_to_modify.item_name.lower():    # if it does
            if item_to_modify.item_price > 0.0:   # if as long as the new item is >t 0.0
                item.item_price = item_to_modify.item_price    # change it
            if item_to_modify.item_quantity > 0:  # if as long as the new item quantity is > 0
                item.item_quantity = item_to_modify.item_quantity # update the quantity
            if item_to_modify.item_description and item_to_modify.item_description.lower() != "none":
                item.item_description = item_to_modify.item_description
            print(f"{item.item_name} has been updated.")
            return # Exit the method once the item is found and changed
    print("Item not found in cart. Nothing to modify")


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.total_cost() for item in self.cart_items)

def print_total(self):
    print(f"{self.customer_name}'s Shopping Cart - {self.current_date}")
    total_items = self.get_num_items_in_cart()
    print(f"Number of Items: {total_items}")
    if total_items == 0:
        print("Your cart is empty")
    else:
            for item in self.cart_items:
                print(f"{item.item_name} {item.item_quantity} @ ${item.item_price} =
${item.total_cost()}")
            print(f"Total: ${self.get_cost_of_cart()}")
```

Module 06: Portfolio Milestone

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

# ------------------------------------------------------------------
# Functions: print_Menu
# ------------------------------------------------------------------
# This function makes the Online shopping cart invoice clear organize and easy to read
def print_Invoice(cart):
    print(f"Customer Cart Name:{cart.customer_name} \t Purchase Date:{cart.current_date}")
    total_quantity = cart.get_num_items_in_cart()
    print("{:>10} {:<10}".format("Total Quantity of Items:", total_quantity))

    print("-" * 85)
    print("{:<10} {:<20} {:<10} {:>15} {:>15}".format("Item", "Description", "Item Quantity", "Cost Per
Unit",
                                                      "Total"))
    print("-" * 85)

    grand_total = 0
    for i, item in enumerate(cart.cart_items):
        total = item.total_cost()
        print("{:<10}{:<30}{:<5} {:>12} {:>18}".format(
            f"Item {i + 1}", item.item_name, item.item_quantity,
             f"${item.item_price:,.2f}",  f"${total:,.2f}"))

        grand_total += total

    tax = grand_total * 0.10
    final_total = grand_total + tax

    print("-" * 85)
    print("{:>70} ${:<10.2f}".format("Total Price test:", grand_total))
    print("{:>70} ${:<10.2f}".format("Tax (10%): ", tax))
    print("{:>70} ${:<10.2f}".format("Total with Tax: ", final_total))
```

Module 06: Portfolio Milestone

```
# The print_menu function
def print_menu(cart):
    while True:
        print("\n" + "+" + "-" * 30 + "+")
        print("|{:^30}|".format(" MENU "))
        print("+" + "-" * 30 + "+")
        print("| a - Add item to cart              |")
        print("| r - Remove item from cart   |")
        print("| c - Change item quantity     |")
        print("| i - Shows item descriptions |")
        print("| o - Output cart invoice          |")
        print("| q - Quit                                |")
        print("+" + "-" * 30 + "+")

        option = input("Select an option:").lower()

        if option == 'a':
            print("Adding items to your cart...")
            print("Type the word *MENU* to return back to the main menu. \n")

            while True:
                name = input("Item Name: ")
                if name.lower() == "menu":
                    break

                description = input("Enter the item description: ")
                if description.lower() == "menu":
                    break
            try:
                price_input = input("Enter the item price: ")
                if price_input.lower() == "menu":
                    break
                price = float(price_input)

                quantity_input = input("Enter the quantity: ")
                if quantity_input.lower() == "menu":
                    break
                quantity = int(quantity_input)
            except ValueError:
                print("Error! Please enter valid numeric values for price and quantity.")
                continue
```

```
            item = ItemToPurchase()
            item.item_name = name
            item.item_description = description
            item.item_price = price
            item.item_quantity = quantity
            cart.add_item(item)
            print("Item added to cart.\n")

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

 elif option == 'c':
    name = input("Enter name of item to modify:")
    try:
            new_price = float(input("Enter new price:"))
            new_quantity = int(input("Enter new quantity: "))
        except ValueError:
            print("Error: Invalid input.")
            continue

        item = ItemToPurchase()
        item.item_name = name
        item.item_price = new_price
        item.item_quantity = new_quantity
        cart.modify_item(item)

elif option == 'i':
    cart.print_description()

elif option == 'o':
    print_Invoice(cart)

elif option == 'q':
    print(" Exit application")
    break
else:
      print("Wrong option. Please choose again.")
```

Module 06: Portfolio Milestone

```
# ---------------------------------------------------------------------
# The Main Program
# ---------------------------------------------------------------------

def main():
    # get user information
    customer_name = input("Customer Shopping Cart Name:")
    current_date = input("Enter Today's date:")
    cart = ShoppingCart(customer_name,current_date)

    print("\n A Shopping Cart Has Been Created For:")
    print(f"Customer: {customer_name}\n Current Date: {current_date}")

    print_menu(cart)

# Run the program.. Finally!
if __name__ == "__main__":
    main()
```

Module 06: Portfolio Milestone

Part II: Screen Shots I : Adding items from the menu selection

```
C:\Users\sarak\AppData\Local\Programs\Python\Python313\python.exe C:\Users\sa
Customer Shopping Cart Name:Barbie M Roberts
Enter Today's date:Febuary 01,2020

A Shopping Cart Has Been Created For:
Customer: Barbie M Roberts
Current Date: Febuary 01,2020

+------------------------------+
|             MENU             |
+------------------------------+
| a - Add item to cart         |
| r - Remove item from cart    |
| c - Change item quantity     |
| i - Shows item descriptions |
| o - Output cart invoice      |
| q - Quit                     |
+------------------------------+
Select an option:a
Adding items to your cart...
Type the word *MENU* to return back to the main menu.

Item Name:Luis Vitton
Enter the item description: Legacy Slingback Pump - Blk - Size 8(US)
Enter the item price: 2500.00
Enter the quantity: 1
Item added to cart.
```

Part II: Screen Shots II : Removing items from the menu selection

```
Item Name:Frito Lays
Enter the item description: Baked Corn Chips -  Family Size - 1 Bag
Enter the item price: 5.50
Enter the quantity: 1
Item added to cart.

Item Name:Menu

+------------------------------+
|             MENU             |
+------------------------------+
| a - Add item to cart         |
| r - Remove item from cart    |
| c - Change item quantity     |
| i - Shows item descriptions |
| o - Output cart invoice      |
| q - Quit                     |
+------------------------------+
Select an option:r
Enter the item to remove:Frito Lays
Frito Lays has been removed from the cart.

+------------------------------+
|             MENU             |
+------------------------------+
| a - Add item to cart         |
| r - Remove item from cart    |
| c - Change item quantity     |
| i - Shows item descriptions |
| o - Output cart invoice      |
| q - Quit                     |
+------------------------------+
Select an option:
```

Part II: Screen Shots III: Changing item quantity, description, and price from the menu selection

```
+------------------------------+
|             MENU             |
+------------------------------+
| a - Add item to cart         |
| r - Remove item from cart    |
| c - Change item quantity     |
| i - Shows item descriptions |
| o - Output cart invoice      |
| q - Quit                     |
+------------------------------+
Select an option:c
Enter name of item to modify:Fishnet Babydolls
Enter new price (or 0 to leave unchanged):25.00
Enter new quantity (or 0 to leave unchanged): 10
Enter new description (or type 'none' to leave unchanged):none
Fishnet Babydolls has been updated.
```

Part II: Screen Shots IV: Shows the item description from the menu selection

```
+------------------------------+
|            MENU              |
+------------------------------+
| a - Add item to cart         |
| r - Remove item from cart    |
| c - Change item quantity     |
| i - Shows item descriptions |
| o - Output cart invoice      |
| q - Quit                     |
+------------------------------+
Select an option:i
Barbie M Roberts's Shopping Cart - Febuary 01,2020
Item Descriptions:
Luis Vitton: Legacy Slingback Pump - Blk - Size 8(US)
DIOR: Le Baume - Toile de Jouy Limited Edition Fragrance - PNK - 1.07oz
 Victoria's Secret Lingerie: Sheer Rose Plunge Bodysuit - Valian Pink - Size M(US)
Fishnet Babydolls: Exotic Dancewear "Babydoll" Fishnet Bodystocking - Pink - Size S(US)
```

Part II: Screen Shots V: outputs the invoice of items to purchase from the menu selection.

```
+-----------------------------+
|            MENU             |
+-----------------------------+
| a - Add item to cart        |
| r - Remove item from cart   |
| c - Change item quantity    |
| i - Shows item descriptions |
| o - Output cart invoice     |
| q - Quit                    |
+-----------------------------+
Select an option:o
Customer Cart Name:Barbie M Roberts      Purchase Date:Febuary 01,2020
Total Quantity of Items: 13
--------------------------------------------------------------------------------
Item       Description          Item Quantity   Cost Per Unit           Total
--------------------------------------------------------------------------------
Item 1    Luis Vitton                   1        $2,500.00          $2,500.00
Item 2    DIOR                          1          $950.00            $950.00
Item 3     Victoria's Secret Lingerie   1          $275.00            $275.00
Item 4    Fishnet Babydolls             10          $25.00            $250.00
--------------------------------------------------------------------------------
                                                Total Price test: $3975.00
                                                      Tax (10%):  $397.50
                                                 Total with Tax:  $4372.50
```

Part II: Screen Shots VI: Exits the application

Leaving the application

```
+-----------------------------+
|            MENU             |
+-----------------------------+
| a - Add item to cart        |
| r - Remove item from cart   |
| c - Change item quantity    |
| i - Shows item descriptions |
| o - Output cart invoice     |
| q - Quit                    |
+-----------------------------+
Select an option:q
 Exit application
```