## **ONLINE SHOPPING CART PROJECT by Joshua Farrell Online Shopping Cart Project Pseudocode** Step 1: Define the ItemToPurchase Class - Attributes: - item\_name (string): Default = "none". - item\_price (float): Default = 0. - item\_quantity (int): Default = 0. - Methods: - Constructor: Initialize attributes with provided or default values. - print\_item\_cost(): - Calculate total cost as item\_quantity \* item\_price. - Print: "item\_name item\_quantity @ $item\_price = $total\_cost".

## Step 2: Create and Prompt for Two Items - In main section of code: - Prompt user for details of Item 1 (name, price, quantity). - Create an ItemToPurchase object for Item 1. - Prompt user for details of Item 2 (name, price, quantity). - Create an ItemToPurchase object for Item 2.

## Step 3: Output Total Cost - Print details for Item 1 and Item 2 using print\_item\_cost(). - Calculate and print the total cost of both items.

**Code:**

## 

## 

## 

## **Executed Code:**

Step 4: Define the ShoppingCart Class  
- Attributes:  
 - customer\_name (string): Default = "none".  
 - current\_date (string): Default = "January 1, 2020".  
 - cart\_items (list): Stores ItemToPurchase objects.  
- Methods:  
 - Constructor: Initialize attributes with provided or default values.  
 - add\_item(item):  
 - Append item to cart\_items.  
 - remove\_item(item\_name):  
 - Search cart\_items for item\_name.  
 - If found, remove item.  
 - If not found, print "Item not found in cart. Nothing removed."  
 - modify\_item(modified\_item):  
 - Search cart\_items for modified\_item.item\_name.  
 - If found, update attributes (if provided).  
 - If not found, print "Item not found in cart. Nothing modified."  
 - get\_num\_items\_in\_cart():  
 - Return sum of item quantities in cart\_items.  
 - get\_cost\_of\_cart():  
 - Return sum of (item\_price \* item\_quantity) for all items in cart\_items.  
 - print\_total():  
 - Print customer\_name and current\_date.  
 - If cart is empty, print "SHOPPING CART IS EMPTY".  
 - Otherwise, print details of each item and the total cost.  
 - print\_descriptions():  
 - Print customer\_name and current\_date.  
 - For each item, print item\_name and item\_description.  
  
Step 5: Implement Menu Functionality  
- Define print\_menu(cart).  
- Display 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 program.  
- Use a loop to handle user input:  
 - Case 'a': Prompt for item details, create an ItemToPurchase object, and call add\_item(item).  
 - Case 'r': Prompt for item name and call remove\_item(item\_name).  
 - Case 'c': Prompt for item details and call modify\_item(modified\_item).  
 - Case 'i': Call print\_descriptions().  
 - Case 'o': Call print\_total().  
 - Default: Print "Invalid option. Please try again."  
  
Step 6: Implement Output Shopping Cart and Item Descriptions  
- OUTPUT SHOPPING CART:  
 - Call print\_total().  
- OUTPUT ITEMS' DESCRIPTIONS:  
 - Call print\_descriptions().

**Code:**

## 

## 

**Executed Code:**

## 

## 

## Step 7: Prompt for Customer Name and Date - In main section: - Prompt user for customer\_name and current\_date. - Create a ShoppingCart object with provided details. - Print customer\_name and current\_date. Step 8: Add Item to Cart - Implement menu option to: - Prompt user for item details (name, description, price, quantity). - Create an ItemToPurchase object and call add\_item(item). Step 9: Remove Item from Cart - Implement menu option to: - Prompt user for item name. - Call remove\_item(item\_name). Step 10: Change Item Quantity - Implement menu option to: - Prompt user for item details (name, new quantity). - Create a modified ItemToPurchase object and call modify\_item(modified\_item).

**Code:**

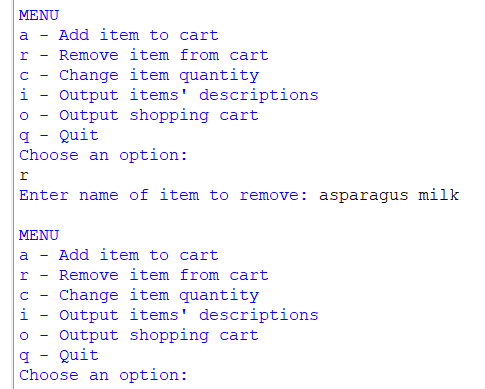
## 

## 

## 

**Executed Code:**

## 



## 

## 