Virsual Programming

**Shopping Card Project**

**Project submission by**

* Muhammad Arbab
* Muhammad Javed
* Muhammad Muzamil

**Submit to**

ATKA ALI

**1. Class: Item.**

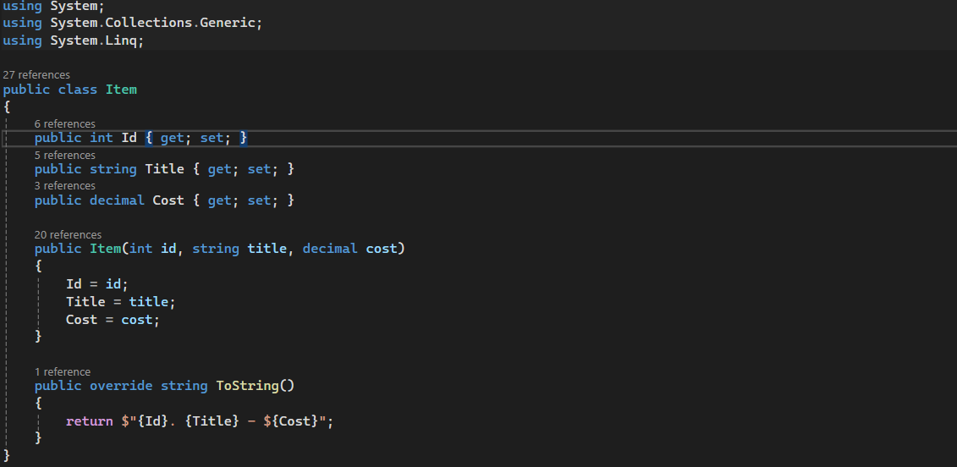
** Purpose:** This shopping cart project is a basic console application developed in C#. The project allows users to add, view, and remove products from a shopping cart, manage quantities, apply discounts and taxes, and complete a purchase.

**Properties**:

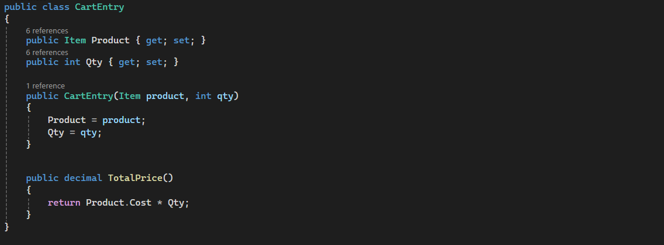
* **Id:** An integer representing the unique identifier for each item.
* **Title:** A string representing the name of the item.
* **Cost:** A decimal representing the price of the item.

**Methods**:

* **ToString():**
  + Returns a formatted string of the item’s ID, title, and cost. This method is useful for displaying item details in the catalog.



2. **Class: CartEntry.**

* **Purpose**:
  + Represents a single entry in the shopping cart, associating an item with a quantity.
* **Properties**:
  + **Product:** An Item object representing the product added to the cart.
  + **Qty:** An integer indicating the quantity of the product in the cart.
* **Methods**:
  + **TotalPrice():** Returns the total price of the cart entry by multiplying the product cost by its quantity. This method aids in calculating the cart’s subtotal.

3. **Class: Cart.**

 **Purpose**:

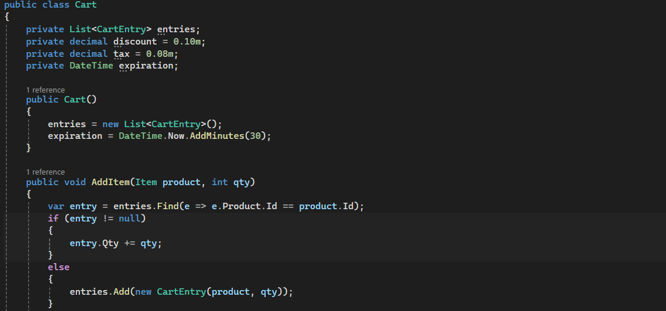
* Manages the shopping cart, including adding and removing items, calculating totals, and handling checkout.

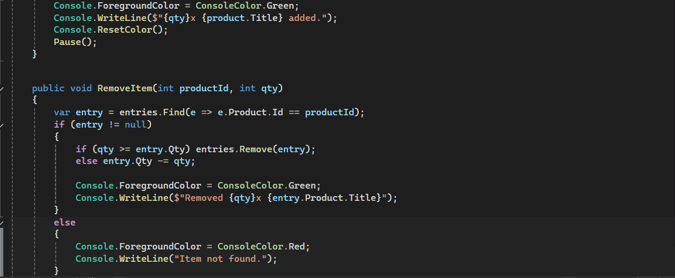
 **Properties**:

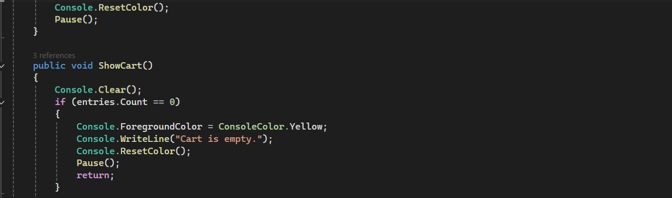
* **entries:** A list of CartEntry objects representing all items in the cart.
* **discount:** A decimal representing a discount rate (10%).
* **tax:** A decimal representing the tax rate (8%).
* **expiration:** A DateTime object representing the cart’s expiration time (30 minutes from creation).

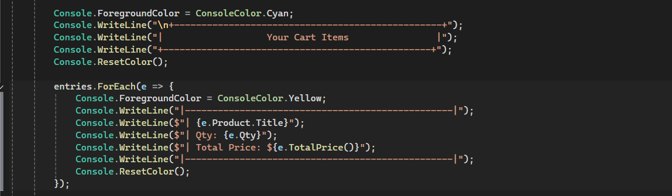
 **Methods**:

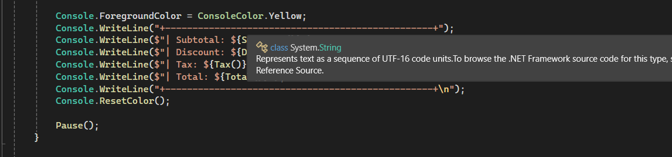
* **AddItem(Item product, int qty):**
  + Adds a product to the cart with a specified quantity. If the product already exists in the cart, it updates the quantity.
* **RemoveItem(int productId, int qty):**
  + Removes a specified quantity of a product from the cart. If the quantity to remove equals or exceeds the current quantity, the item is removed completely.
* **ShowCart():**
  + Displays the current cart items with their quantities, prices, subtotal, discount, tax, and total.
* **Subtotal():**
  + Calculates and returns the subtotal by summing up the total prices of all cart entries.
* **Discount():** Calculates and returns the discount applied to the subtotal.
* **Tax():** Calculates and returns the tax based on the subtotal after discount.
* **Total():** Calculates and returns the final total, including discount and tax.
* **IsExpired():** Checks if the cart session has expired by comparing the current time with the expiration time.
* **CompletePurchase():** Completes the checkout process by displaying the cart summary, clearing entries, and thanking the user if the cart has not expired.
* **Pause():** Pauses execution and waits for the user to press Enter to continue. Used after displaying messages for smoother navigation.

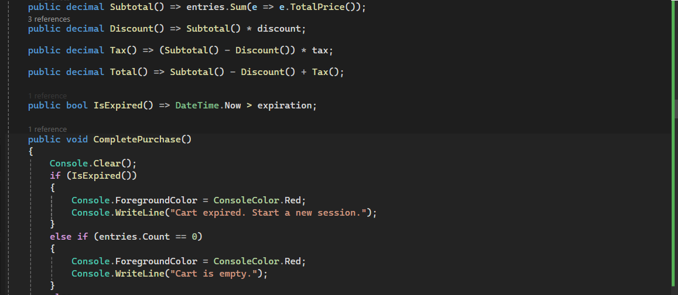


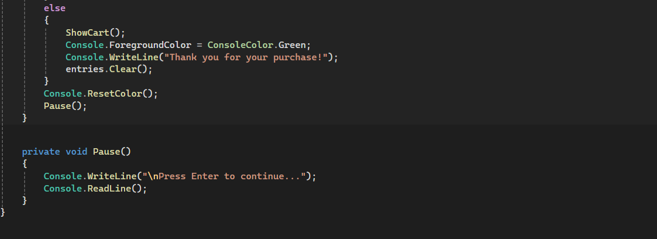










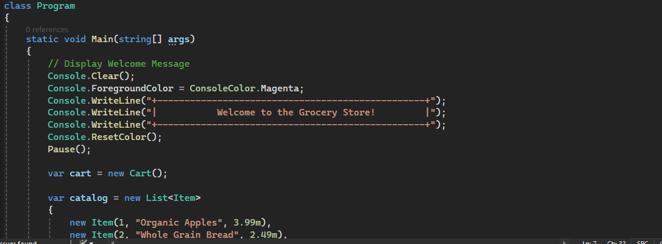


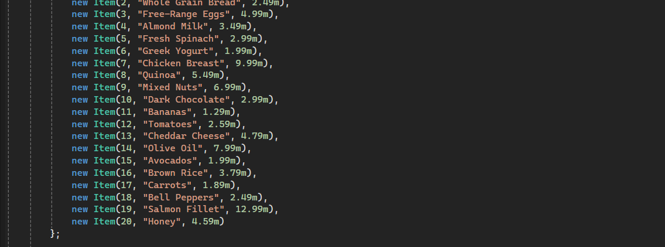
4. **Class: Program.**

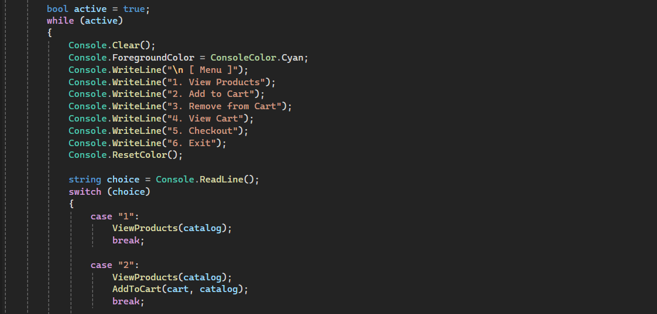
 **Purpose**: The main entry point for the application. It initializes the catalog, displays the main menu, and manages user interactions.

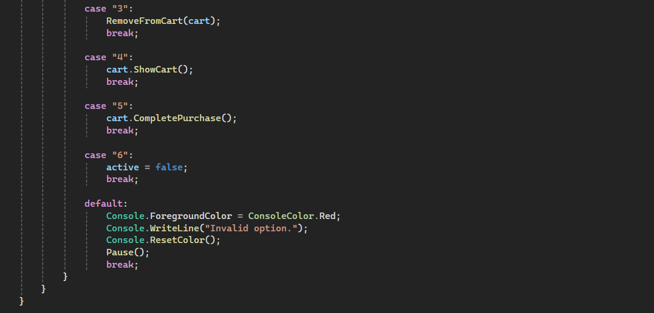
 **Methods**:

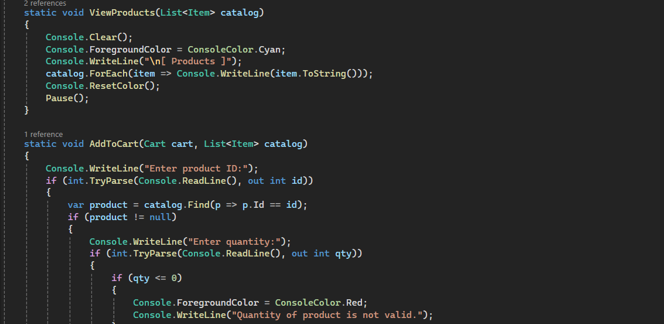
* **Main():** Initializes the application and handles the main program flow. Displays the main menu, takes user input, and directs the user to various actions based on their selection.
* **ViewProducts(List<Item> catalog):** Displays all items in the catalog. Used to list available products for the user.
* **AddToCart(Cart cart, List<Item> catalog):** Prompts the user to add a product to the cart by entering a product ID and quantity. Validates inputs and adds the item to the cart if valid.
* **RemoveFromCart(Cart cart):** Allows the user to remove a specific quantity of an item from the cart by entering the item ID and quantity.
* **Pause():** Pauses execution and waits for the user to press Enter to continue, facilitating smooth user navigation.

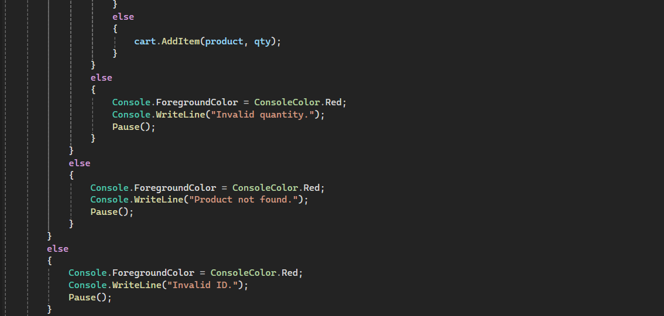
****

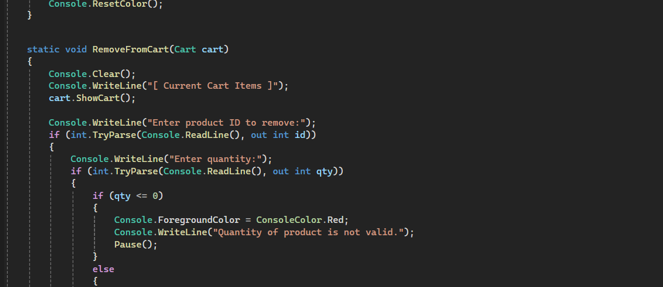
****

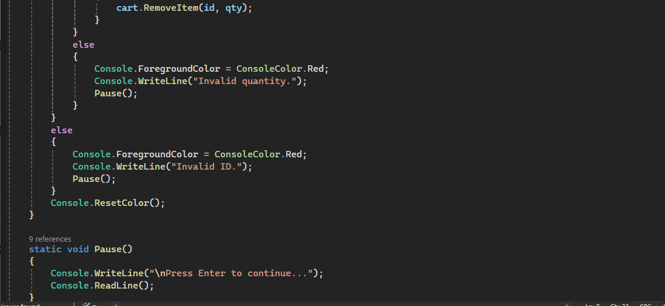
****











**UML DIAGRAM**

