Assignment 4: E-Commerce System

Topics Covered:

- Classes and objects
- Inheritance
- Polymorphism
- Encapsulation
- C# syntax essentials
- Interface implementation
- Method overloading
- Composition

Objective: Create a comprehensive e-commerce system that demonstrates advanced OOP concepts, including interfaces, composition, and more complex class hierarchies.

Duration: 120-150 minutes

Requirements:

1. Create an `IDiscountable` interface with:
    - Method `ApplyDiscount(decimal percentage)` that returns the discounted price
    - Property `IsDiscountable` (boolean)
2. Create an abstract `Product` class that implements `IDiscountable`:
    - Private fields for product ID, name, description, price, and inventory count
    - Properties with appropriate access modifiers
    - Constructor with parameter validation
    - Abstract method `CalculateTax()` that returns the tax amount
    - Virtual method `GetDetails()` that returns a string with product information
    - Implementation of `IDiscountable` methods
3. Create at least three derived product classes (e.g., `ElectronicProduct`, `ClothingProduct`, `BookProduct`) that:
    - Implement the abstract methods differently (e.g., different tax rates)
    - Add unique properties (e.g., warranty period, size, author)
    - Override `GetDetails()` to include specific information
4. Create a `ShoppingCart` class that:
    - Uses composition to store a collection of products and quantities
    - Has methods to add and remove products, update quantities
    - Calculates subtotal, tax, and total price
    - Applies discounts to eligible products
5. Create a `Customer` class with:
    - Personal information (name, email, address)

  - Shopping cart as a property (composition)
  - Membership level (enum: Regular, Premium, VIP)
  - Method to calculate loyalty discounts based on membership level

6. Create an `Order` class that:

   - Has a unique order ID
   - Stores customer information, order date, and products purchased
   - Has different shipping methods (Standard, Express, Next-Day)
   - Calculates shipping costs based on product weight and shipping method
   - Generates order summary and receipt

7. Create an `ECommerceSystem` class that:

   - Maintains inventory of products
   - Processes orders
   - Handles customer management
   - Generates sales reports

8. Implement exception handling for:

   - Out-of-stock products
   - Invalid discount codes
   - Payment processing errors

Sample Input:

```
// Create products
ElectronicProduct laptop = new ElectronicProduct("E001", "Gaming Laptop",
1200.00m, 10);
laptop.WarrantyPeriod = 24; // months
laptop.Weight = 2.5; // kg

BookProduct book = new BookProduct("B001", "C# Programming Guide", 45.99m, 50);
book.Author = "John Smith";
book.ISBN = "978-3-16-148410-0";
book.Weight = 0.8; // kg

ClothingProduct shirt = new ClothingProduct("C001", "Casual Shirt", 29.99m, 100);
shirt.Size = "L";
shirt.Color = "Blue";
shirt.Weight = 0.3; // kg

// Create customer
Customer customer = new Customer("Alice Johnson", "alice@example.com");
customer.Address = "123 Main St, Anytown, USA";
customer.MembershipLevel = MembershipLevel.Premium;

// Add products to shopping cart
customer.ShoppingCart.AddProduct(laptop, 1);
customer.ShoppingCart.AddProduct(book, 2);
customer.ShoppingCart.AddProduct(shirt, 3);
```

```
// Apply discounts
laptop.ApplyDiscount(10.0m); // 10% discount on electronics
book.ApplyDiscount(5.0m);    // 5% discount on books

// Calculate loyalty discount
decimal loyaltyDiscount = customer.CalculateLoyaltyDiscount();

// Create order
Order order = new Order(customer, "ORD-2023-001");
order.ShippingMethod = ShippingMethod.Express;
order.ProcessOrder();

// Generate receipt
string receipt = order.GenerateReceipt();
Console.WriteLine(receipt);

// Try to order an out-of-stock product
try
{
    ElectronicProduct smartphone = new ElectronicProduct("E002", "Smartphone", 800.00m, 0);
    customer.ShoppingCart.AddProduct(smartphone, 1);
}
catch (Exception ex)
{
    Console.WriteLine($"Error: {ex.Message}");
}
```

Sample Output:

```
====== ORDER RECEIPT ======
Order ID: ORD-2023-001
Date: 2023-05-15 14:30:22

Customer: Alice Johnson
Email: alice@example.com
Address: 123 Main St, Anytown, USA
Membership: Premium

Items:
1. Gaming Laptop (E001)
   Original Price: $1,200.00
   Discount: $120.00 (10%)
   Quantity: 1
   Subtotal: $1,080.00
   Tax (8.25%): $89.10

2. C# Programming Guide (B001)
   Original Price: $45.99
   Discount: $2.30 (5%)
   Quantity: 2
   Subtotal: $87.38
   Tax (6.00%): $5.24
   Details: Author: John Smith, ISBN: 978-3-16-148410-0

3. Casual Shirt (C001)
   Original Price: $29.99
   Discount: $0.00 (0%)
   Quantity: 3
   Subtotal: $89.97
   Tax (7.50%): $6.75
   Details: Size: L, Color: Blue

Order Summary:
Subtotal: $1,257.35
Tax: $101.09
Loyalty Discount: $62.87 (5% Premium membership)
Shipping: $15.00 (Express)
Total: $1,310.57

Thank you for shopping with us!
============================

Error: Product 'Smartphone' (E002) is out of stock.
```