

**VGP133 Intro to C# Programming**

**Assignment 3**

Advanced Classes and OOP in C#

*Estimated hours:​ ~4 hours*

**

Assignment Goals:

* Apply and practice some basic concepts in C# related to classes and objects
* Practice working with multiple files and scripts in C#

Assignment Hand-in

* Please upload the C# project on **Omnivox**

Assignment Overview

Please answer the following questions. Make sure that all of them are written in C# using the Visual Studio 2022 IDE. Make sure that all properties (aka data members) are private, and the classes have their respective setters and getters if necessary.

In this assignment, you’ll be designing and implementing an Online Shopping System.

1. **Inheritance (10 pts)**
2. Product Class

Create a base class named **Product** to represent products that can be purchased. Include properties **ProductName, Price, Discount, and Description**.

Then, create methods with the following functionalities:

**CalculateTax** – receives a float on what rate it’s taxed and returns the result.

**GetPrice** – returns a float which calculates the cost of the product, including the discount.

**DisplayInfo** – For the base class version, display all the properties contained in the product class.

1. Book Class

Create a derived class named Book that will inherit from the Product base class. Include properties **Author, Genre, and PublicationYear**.

Methods:

**DisplayInfo** – override the base class method to display book-specific information including author, genre, and publication year.

**IsModern** – returns whether the book was published in the year 2015 onwards or not.

1. Electronics Class

Create a derived class named **Electronics** that will inherit from the Product base class. Include properties **Brand, Model**, and **WarrantyPeriod (in years)**.

Methods:

**DisplayInfo** – override the base class method to display electronic-specific information including brand, model, and warranty period.

**IsWarrantyValid** – receives a float for age of the electronic and then returns a bool. The method returns true if the warranty is still valid, and false if it’s past the year.

1. Clothing Class

Create a derived class named **Clothing** that will inherit from the Product base class. Include properties **Size, Color, and Material**.

Methods:

**DisplayInfo** – override the base class method to display electronic-specific information including brand, model, and warranty period.

**IsAvailableInColor** – receives a string, and checks if the color is available. Returns a true if it’s available, and false if it’s not.

1. **Polymorphism (5 pts)**

Implement a method named **DisplayInfo** in the base class **Product**. This method should be overridden in each derived class to display product-specific information.

Create a collection (e.g., a list or array) of different products from various categories. Demonstrate polymorphism by iterating through the collection and calling the DisplayInfo method on each product to display their information

1. **Abstraction (10 pts)**

Define an abstract class called **PaymentMethod** with common properties such as **PaymentID** and **PaymentAmount**. Include an abstract method named **ProcessPayment** that must be implemented by any derived class.

Create derived classes for different payment methods (e.g., **CreditCard, PayPal, and Cash**) that are inherited from the PaymentMethod abstract class. Implement the ProcessPayment method in each concrete class according to the specific payment process

1. **Interfaces (10 pts)**

Define an interface named **IShoppingCart** with methods for adding products to the cart, removing products, calculating the total price, and checking out.

Implement the **IShoppingCart** interface in a **ShoppingCart** class. The **ShoppingCart** class should include methods for managing the shopping cart (add and remove products), calculating total price with applying discounts, and completing the purchase by checking out.

1. **Main Program (10 pts)**

Create a program that simulates an online shopping experience with the following functionalities.

* Display all the available products that can be purchased.
* Add Products to Shopping Cart
* Finish Shopping and Checkout the Product.

Make sure to have proper error handling, and user-friendly messages.