**C# Abstraction and Inheritance Activity: Shopping Cart Scenario**

**Question:**

Imagine you are developing an e-commerce application in C#. One of the features you need to implement is a shopping cart system. The system allows users to add, remove, and view products, and calculate the total price based on the products in the cart. You need to design this system using **Abstraction** and **Inheritance** principles.

**Part 1: Abstraction**

* Create an abstract class Product that represents common properties and methods for different types of products.
  + Include a property Price (type decimal) that represents the price of the product.
  + Include an abstract method DisplayInfo() that will be implemented by derived classes to show product details.
* Create two derived classes:
  + ElectronicProduct that includes an additional property WarrantyPeriod (type int to represent the warranty period in months) and overrides the DisplayInfo() method.
  + GroceryProduct that includes an additional property ExpirationDate (type DateTime) and overrides the DisplayInfo() method.

**Part 2: Inheritance**

* Create a class ShoppingCart that can hold multiple products.
  + The class should have methods to:
    - AddProduct(Product product): Add a product to the cart.
    - RemoveProduct(Product product): Remove a product from the cart.
    - CalculateTotalPrice(): Calculate and return the total price of all products in the cart.
    - DisplayCartContents(): Display the details of all products in the cart using the DisplayInfo() method of each product.
* In the Main method, demonstrate the use of the ShoppingCart class with both ElectronicProduct and GroceryProduct objects. Show how inheritance and abstraction are applied by displaying the product details and the total price.