Quiz#2

For the inventory management system of a retail store, We'll start with a base class named "Product" to store common attributes like product name, price, and quantity. Then, we'll create specialized classes like "Electronics" and "Clothing" that inherit from the base class to manage specific types of products. To manage memory efficiently, we'll use dynamic memory allocation.

To calculate profits, we'll create a friend function called "Profit_Calculator" that can access private data of the Product class. Polymorphism will allow us to customize how product details are displayed using virtual functions. And with operator overloading, we can compare products based on their prices or categories easily.

Class Names, Function Names, and Attribute Names:

1. Class Name: **Product**
    - Attributes:
        - productName (string)
        - productID (int)
        - price (float)
        - quantityAvailable (int)
    - Functions:
        - virtual void displayDetails(): Displays details of the product.
        - virtual float calculateProfit(): Calculates the profit generated from sales.
2. Derived Class Name: **Electronics**
    - Attributes:
        - warrantyPeriod (int)
        - powerConsumption (float)
    - Functions:
        - void displayDetails(): Overrides the base class function to display electronics-specific details.
3. Derived Class Name: **Clothing**

- Attributes:
    - size (string)
    - material (string)
- Functions:
    - void displayDetails(): Overrides the base class function to display clothing-specific details.

4. Function Name: **Profit_Calculator**
    - Function Type: Friend Function
    - Parameters:
        - product (Product object)
        - salesQuantity (int)
    - Returns:
        - float (profit generated from sales)
5. Operator Overloading:
    - Operator: **<** (less than)
    - Usage: Enables comparison of products based on Profit.