Prelim Exam II Phase: Inventory Management

Objective: The goal is to create a program that manages an inventory system using multi-dimensional arrays. The program will store product details (including names and stock quantities), display them to the user, allow updates to the inventory stock, and provide an option to reset the inventory to its initial values.

⚠️ The program must apply separation of concerns by creating:

- **InventoryService** – handles business logic and data
- **InventoryView** – handles user interaction and display

Program.cs should only create and run the InventoryView.

Implementation Steps:

1. Create a new C# Console Application.

○ Set up a new C# console application in Visual Studio or any C# IDE of your choice.

○ Add two folders:
■ **Services** – create **InventoryService.cs**
■ **Views** – create **InventoryView.cs**

---

2. Define a Two-Dimensional Array to Store Product Details (Inside InventoryService).

○ The two-dimensional array will store product information across two rows:

■ Row 1 will store product names (e.g., Apples, Milk, Bread).
■ Row 2 will store the corresponding stock quantities for each product (e.g., 10, 5, 20).

○ Array Structure:
■ **products[0, 0] = "Apples"** — The first row stores the product names.

■ **products[1, 0] = "10"** — The second row stores the stock quantities for the corresponding product in the first row.

○ Also create an **initialStock** array to store the original stock values for reset purposes.

⚠️ InventoryService should contain the array and methods for:
■ **Viewing inventory data**
■ **Updating stock**
■ **Resetting inventory**

It should NOT handle console **input/output.**

---

3. Display Inventory Details (Inside InventoryView).

○ The program will display the inventory in a structured format, showing product names along with their stock quantities.

○ This will be done by calling methods from InventoryService and iterating through the returned data.

○ Create a **public Run() method** inside **InventoryView**. This method will control the main program loop and handle menu selection.

⚠️ InventoryView handles all user interaction and console display.

---

4. Create a Menu with Options (Inside InventoryView).

○ **Option 1: View Inventory** – This option will display the current inventory, listing all products along with their respective stock quantities.

○ **Option 2: Update Stock** – This option will allow the user to update the stock for a specific product by selecting the product by number (1, 2, or 3). The user can enter a new stock quantity for the selected product.

○ **Option 3: Reset Inventory** – This new option will allow the user to reset all products' stock quantities to their original values (as defined in the `initialStock` array).

○ **Option 4: Exit** – This option will allow the user to exit the program.

---

5. Keep the Program Running

○ Use a loop inside **InventoryView** to keep the program running until the user selects the "Exit" option.

○ The loop will continuously display the menu options and perform the chosen actions.

---

6. Program Entry Point (Program.cs)

○ Create only one instance of **InventoryView.**
○ Call the **Run()** method of **InventoryView.**

⚠️ **Program.cs** should NOT contain inventory logic or menu handling.