July 2025 CSE 108

Practice Problem

You need to simulate an **Inventory Management System** for an online store using basic Object-Oriented Programming (OOP) concepts in C++. The system should allow the store manager to maintain a collection of products, track stock updates (both additions and reductions), and retrieve the product with the highest current stock. For each product, a maximum of **100 stock change records** can be kept due to resource limitations. *For storing strings, you can use the C++ <string> library if convenient.*

Tasks

1. Create a class named `Product` with the following private member variables:
    - `string productName` – name of the product
    - `string productID` – a unique identifier for the product
    - `double price` – price of the product
    - `int stockChanges[100]` – statically allocated array of fixed length 100 to store stock additions/reductions
    - `int numStockChanges` – current number of stock updates
2. Include public member functions to:
    - Add stock: `void addStock(int amount)`
    - Reduce stock: `void reduceStock(int amount)`
    - Get current stock: `int getCurrentStock()`
    - Display product details: `void display()`
3. Implement the following for the `Product` class:
    - A default constructor
    - A parameterized constructor: `Product(string productName, string productID, double price)`

4. Create a class named `Inventory` to manage a collection of `Product` objects.
5. Include private member variables:
    - `Product* products` – dynamically allocated array of `Product`
    - `int numProducts` – current number of products in the inventory
    - `int maxProducts` – maximum number of products that can be stored
6. Include public member functions to:
    - Add a new product to the inventory: `void addProduct(Product p)`
    - Update stock (add/reduce) of a specific product by ID: `void updateProductStock(string productID, int amount)`
        - If the amount is positive, add stock; if negative, reduce stock.
    - List all products in the inventory: `void listAllProducts()`
    - Return the product with the highest current stock: `Product getMostStockedProduct()`
        - If multiple products have the highest stock, return the one with the **fewest stock changes**. If still tied, return any.
7. Implement the following for the `Inventory` class:
    - A default constructor
    - Parameterized constructors:
        - `Inventory(int maxProducts)`
        - `Inventory(Product* products, int numProducts, int maxProducts)`
    - A copy constructor
    - A destructor
8. Please carefully see the sample main function to understand the output format. You can copy the main function directly to your code.

Sample `main()` Function:

```
int main() {
    Inventory inv1(5);
    Product p1("Laptop", "P001", 50000);
    Product p2("Smartphone", "P002", 30000);
    Product p3("Tablet", "P003", 25000);
    Product p4("Smartwatch", "P004", 4000);
    Product p5("Headphones", "P005", 2000);
    Product p6("Charger", "P006", 1000);

    inv1.addProduct(p1);
    inv1.addProduct(p2);
    inv1.addProduct(p3);
    inv1.addProduct(p4);
    inv1.addProduct(p5);
    inv1.addProduct(p6);

    inv1.updateProductStock("P001", 50);
    inv1.updateProductStock("P001", -10);
    inv1.updateProductStock("P003", 70);
    inv1.updateProductStock("P002", 70);
    inv1.updateProductStock("P002", -5);
    inv1.updateProductStock("P003", -10);

    cout << "\nProducts in Inventory 1:\n";
    inv1.listAllProducts();
    cout << "\nMost Stocked Product in Inventory 1:\n";
    inv1.getMostStockedProduct().display();

    Inventory inv2 = inv1;
    inv2.updateProductStock("P003", 10);
    inv2.updateProductStock("P002", 10);
    inv2.updateProductStock("P002", -5);

    cout << "\nProducts in Inventory 2:\n";
    inv2.listAllProducts();

    cout << "\nMost Stocked Product in Inventory 1:\n";
    inv1.getMostStockedProduct().display();

    cout << "\nMost Stocked Product in Inventory 2:\n";
    inv2.getMostStockedProduct().display();
    return 0;
}
```

Expected Output:

```
Product added with ID: P001
Product added with ID: P002
Product added with ID: P003
Product added with ID: P004
Product added with ID: P005
Inventory full. Cannot add any more product.

Products in Inventory 1:
Product: Laptop (ID: P001), Price: Tk.50000, Stock: 40
Product: Smartphone (ID: P002), Price: Tk.30000, Stock: 65
Product: Tablet (ID: P003), Price: Tk.25000, Stock: 60
Product: Smartwatch (ID: P004), Price: Tk.4000, Stock: 0
Product: Headphones (ID: P005), Price: Tk.2000, Stock: 0

Most Stocked Product in Inventory 1:
Product: Smartphone (ID: P002), Price: Tk.30000, Stock: 65

Products in Inventory 2:
Product: Laptop (ID: P001), Price: Tk.50000, Stock: 40
Product: Smartphone (ID: P002), Price: Tk.30000, Stock: 70
Product: Tablet (ID: P003), Price: Tk.25000, Stock: 70
Product: Smartwatch (ID: P004), Price: Tk.4000, Stock: 0
Product: Headphones (ID: P005), Price: Tk.2000, Stock: 0

Most Stocked Product in Inventory 1:
Product: Smartphone (ID: P002), Price: Tk.30000, Stock: 65

Most Stocked Product in Inventory 2:
Product: Tablet (ID: P003), Price: Tk.25000, Stock: 70
```