**Hafiza Amber Fatima**

**1.Product Management**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**class Product {**

**public:**

**int id;**

**string name;**

**double price;**

**string description;**

**int stock;**

**void display() {**

**cout << "ID: " << id**

**<< ", Name: " << name**

**<< ", Price: $" << price**

**<< ", Description: " << description**

**<< ", In Stock: " << stock << "\n";**

**}**

**};**

**class ProductManager {**

**Product productList[100];**

**return; int totalProducts = 0;**

**public:**

**void addProduct() {**

**if (totalProducts >= 100) {**

**cout << "Product limit reached. Cannot add more.\n";**

**}**

**Product newProduct;**

**cout << "Enter Product ID: ";**

**cin >> newProduct.id;**

**cin.ignore();**

**cout << "Enter Product Name: ";**

**getline(cin, newProduct.name);**

**cout << "Enter Product Price: ";**

**cin >> newProduct.price;**

**cin.ignore();**

**cout << "Enter Product Description: ";**

**getline(cin, newProduct.description);**

**cout << "Enter Quantity in Stock: ";**

**cin >> newProduct.stock;**

**productList[totalProducts++] = newProduct;**

**cout << "Product added successfully!\n";**

**}**

**void updateProduct() {**

**int id;**

**cout << "Enter Product ID to update: ";**

**cin >> id;**

**for (int i = 0; i < totalProducts; i++) {**

**if (productList[i].id == id) {**

**cin.ignore();**

**cout << "Enter New Name: ";**

**getline(cin, productList[i].name);**

**cout << "Enter New Price: ";**

**cin >> productList[i].price;**

**cin.ignore();**

**cout << "Enter New Description: ";**

**getline(cin, productList[i].description);**

**cout << "Enter New Stock Quantity: ";**

**cin >> productList[i].stock;**

**cout << "Product updated successfully!\n";**

**return;**

**}**

**}**

**cout << "Product not found.\n";**

**}**

**[j + 1];**

**}**

**void deleteProduct() {**

**int id;**

**cout << "Enter Product ID to delete: ";**

**cin >> id;**

**for (int i = 0; i < totalProducts; i++) {**

**if (productList[i].id == id) {**

**for (int j = i; j < totalProducts - 1; j++) {**

**productList[j] = productList totalProducts--;**

**cout << "Product deleted successfully.\n";**

**return;**

**}**

**}**

**cout << "Product not found.\n";**

**}**

**void viewProducts() {**

**if (totalProducts == 0) {**

**cout << "No products available.\n";**

**return;**

**}**

**for (int i = 0; i < totalProducts; i++) {**

**productList[i].display();**

**}**

**}**

**Product\* getProductById(int id) {**

**for (int i = 0; i < totalProducts; i++) {**

**if (productList[i].id == id) {**

**return &productList[i];**

**}**

**}**

**return nullptr;**

**}**

**};**

**Description**

**This C++ code defines a simple product management system with two main classes: Product and ProductManager.**

**1. Product Class**

**This class represents a single product with the following attributes:**

* **id: An integer representing the product ID.**
* **name: A string for the product name.**
* **price: A double for the product price.**
* **description: A string describing the product.**
* **stock: An integer for quantity available in stock.**

**It includes one method:**

* **display(): Outputs the product's details to the console.**

**---**

**2. ProductManager Class**

**This class manages a collection of Product objects using an array (productList) with a maximum capacity of 100 products. It tracks the number of products using totalProducts.**

**It provides the following methods:**

* **addProduct():**

**Prompts the user to enter the details of a new product and adds it to the list. It checks whether the list has reached its limit.**

* **updateProduct():**

**Allows the user to update the details of an existing product by providing its ID. If the product is found, it updates its attributes.**

* **deleteProduct():**

**Removes a product from the list by its ID. It shifts the remaining elements to maintain the order.**

* **viewProducts():**

**Displays the details of all the products. If no products exist, it shows a corresponding message.**

* **getProductById(int id):**

**Returns a pointer to the product matching the given ID. If not found, it returns nullptr.**

**---**

**Overall Purpose**

**This code is a basic implementation of a console-based inventory management system, allowing users to:**

**Add, update, delete, and view products**

**Search for products by ID**

**It's useful for beginner-level projects or learning object-oriented programming concepts in C++.**