The University of Lahore, CS & IT Department

Object Oriented Programming

Assignment # 04

Start Date: 03/1/2025 **Total Marks: 10**

Due Date: 10/1/2025 **Program:** BSCS

Instructions

1. Understanding the problem is part of the assignment. So, no query, please.
2. You will get zero marks if found any type of plagiarism or AI generated code.
3. No submission after due date.
4. Upload pdf file containing your code and screenshots of outputs.

Name: Arslan Fayyaz

Sap id: 70149152

Section: (A)

Subject: (OOP)

Department: (Computer Science &IT)

Submitted To: Mam Mishal Muneer

Question – 01: Plant Nursery Inventory Management using "File Handling"

Solution:

```
#include <iostream>
```

```
#include <fstream>
#include<iomanip>
#include<vector>
#include <sstream>



using namespace std;

class Plant {
public:
    int plantID;
    string name;
    string category;
    float price;
    int quantity;


    Plant(int id, string n, string c, float p, int q)
        : plantID(id), name(n), category(c), price(p), quantity(q) {}
```

```
    void display() const {
        cout << "ID: " << plantID << ", Name: " << name
             << ", Category: " << category << ", Price: $" << fixed << setprecision(2) << price
             << ", Quantity: " << quantity << endl;
    }

    string toFileString() const {
        return to_string(plantID) + "," + name + "," + category + "," + to_string(price) + "," + to_string(quantity);
    }

    void updateQuantity(int newQuantity) {
        quantity = newQuantity;
    }
};

vector<Plant> loadInventoryFromFile() {
    vector<Plant> plantList;
    ifstream file("plant_inventory.txt");
```

```
    if (file.is_open()) {
        string line;
        while (getline(file, line)) {
            stringstream ss(line);
            string id, name, category, price, quantity;

            getline(ss, id, ',');
            getline(ss, name, ',');
            getline(ss, category, ',');
            getline(ss, price, ',');
            getline(ss, quantity, ',');

            plantList.push_back(Plant(stoi(id), name, category, stof(price), stoi(quantity)));
        }
        file.close();
    }
    return plantList;
}

void saveInventoryToFile(const vector<Plant>& plantList) {
    ofstream file("plant_inventory.txt");
```

```
    if (file.is_open()) {
        for (const auto& plant : plantList) {
            file << plant.toFileString() << endl;
        }
        file.close();
        cout << "Inventory saved to file." << endl;
    } else {
        cout << "Error opening file to save inventory." << endl;
    }
}

bool addPlant(vector<Plant>& plantList, int plantID, string name, string category,
float price, int quantity) {

    for (const auto& plant : plantList) {
        if (plant.plantID == plantID || plant.name == name) {
            cout << "Plant already exists! Please add a new plant." << endl;
            return false;
        }
    }

    plantList.push_back(Plant(plantID, name, category, price, quantity));
    cout << "Plant added successfully!" << endl;
```

```
    return true;
}



void viewPlants(const vector<Plant>& plantList) {
    if (plantList.empty()) {
        cout << "The inventory is empty." << endl;
    } else {
        cout << "\n--- Plant Inventory ---\n";
        for (const auto& plant : plantList) {
            plant.display();
            cout << "------------------------" << endl;
        }
    }
}



void updateQuantity(vector<Plant>& plantList, int plantID, int newQuantity) {
    for (auto& plant : plantList) {
        if (plant.plantID == plantID) {
            plant.updateQuantity(newQuantity);
            cout << "Quantity updated successfully!" << endl;
```

```
            saveInventoryToFile(plantList);
            return;
        }
    }
    cout << "Error: Plant with ID " << plantID << " not found." << endl;
}

int main() {
    vector<Plant> plantList = loadInventoryFromFile();

    int choice;
    while (true) {

        cout << "\n--- Plant Inventory System ---\n";
        cout << "1. Add a Plant\n";
        cout << "2. View Plants\n";
        cout << "3. Update Quantity\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        if (choice == 1) {
```

```
    int plantID, quantity;
    string name, category;
    float price;

    cout << "Enter Plant ID: ";
    cin >> plantID;
    cin.ignore();
    cout << "Enter Plant Name: ";
    getline(cin, name);
    cout << "Enter Plant Category: ";
    getline(cin, category);
    cout << "Enter Plant Price: $";
    cin >> price;
    cout << "Enter Plant Quantity: ";
    cin >> quantity;

    addPlant(plantList, plantID, name, category, price, quantity);

} else if (choice == 2) {

    viewPlants(plantList);
```

```
    } else if (choice == 3) {

        int plantID, newQuantity;
        cout << "Enter Plant ID to update: ";
        cin >> plantID;
        cout << "Enter new quantity: ";
        cin >> newQuantity;
        updateQuantity(plantList, plantID, newQuantity);

    } else if (choice == 4) {

        cout << "Exiting the system." << endl;
        break;

    } else
        cout << "Invalid choice. Please try again." << endl;
    }
}
```

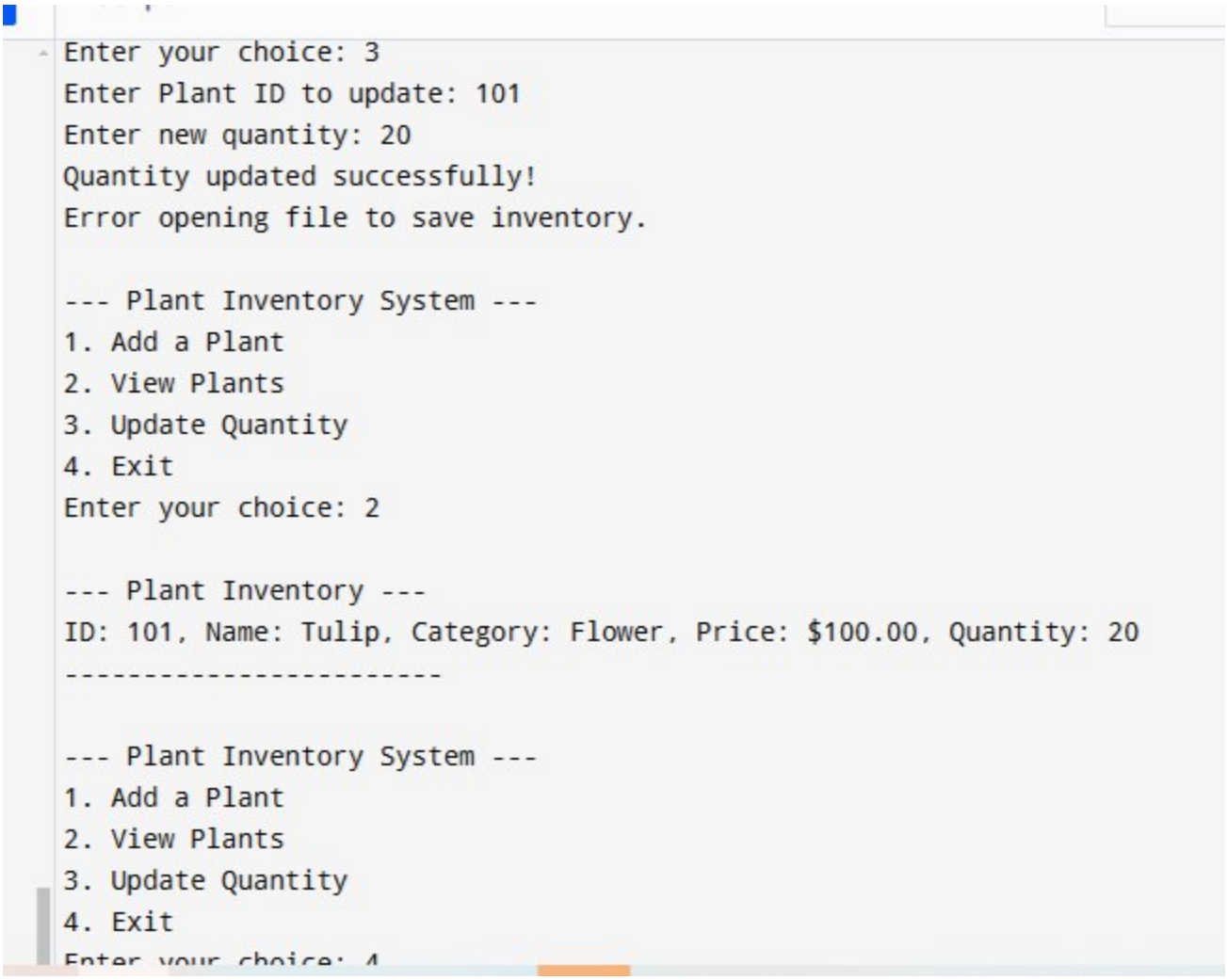
Output:1

```
--- Plant Inventory System ---
1. Add a Plant
2. View Plants
3. Update Quantity
4. Exit
Enter your choice: 1
Enter Plant ID: 101
Enter Plant Name: Tulip
Enter Plant Category: Flower
Enter Plant Price: $100
Enter Plant Quantity: 20
Plant added successfully!

--- Plant Inventory System ---
1. Add a Plant
2. View Plants
3. Update Quantity
4. Exit
Enter your choice: 2

--- Plant Inventory ---
ID: 101, Name: Tulip, Category: Flower, Price: $100.00, Quantity: 20
```

Output no 2:


Enter your choice: 3
Enter Plant ID to update: 101
Enter new quantity: 20
Quantity updated successfully!
Error opening file to save inventory.
--- Plant Inventory System ---
1. Add a Plant
2. View Plants
3. Update Quantity
4. Exit
Enter your choice: 2
--- Plant Inventory ---
ID: 101, Name: Tulip, Category: Flower, Price: $100.00, Quantity: 20

--- Plant Inventory System ---
1. Add a Plant
2. View Plants
3. Update Quantity
4. Exit

Output no 3:

```
Error opening file to save inventory.

--- Plant Inventory System ---
1. Add a Plant
2. View Plants
3. Update Quantity
4. Exit
Enter your choice: 2

--- Plant Inventory ---
ID: 101, Name: Tulip, Category: Flower, Price: $100.00, Quantity: 20
------------------------

--- Plant Inventory System ---
1. Add a Plant
2. View Plants
3. Update Quantity
4. Exit
Enter your choice: 4


=== Code Execution Successful ===
```