**CODE: -**

#include <iostream>

#include <string>

using namespace std;

class InventoryManager {

public:

InventoryManager();

~InventoryManager();

void run();

private:

struct Item {

string name;

int quantity;

double price;

};

Item\* inventory;

int size;

void displayMenu();

void addItem();

void displayItems();

void deleteItem();

};

InventoryManager::InventoryManager() {

inventory = nullptr;

size = 0;

}

InventoryManager::~InventoryManager() {

if (inventory) {

delete[] inventory;

}

}

void InventoryManager::run() {

int choice;

while (true) {

displayMenu();

cin >> choice;

switch (choice) {

case 1:

addItem();

break;

case 2:

displayItems();

break;

case 3:

deleteItem();

break;

case 4:

cout << "Exiting the program." << endl;

return;

default:

cout << "Invalid choice. Please try again." << endl;

}

}

}

void InventoryManager::displayMenu() {

cout << "Inventory Management System" << endl;

cout << "1. Add Item" << endl;

cout << "2. Display Items" << endl;

cout << "3. Delete Item" << endl;

cout << "4. Exit" << endl;

cout << "Enter your choice: ";

}

void InventoryManager::addItem() {

Item newItem;

cout << "Enter the name of the item: ";

cin >> newItem.name;

cout << "Enter the quantity: ";

cin >> newItem.quantity;

cout << "Enter the price: ";

cin >> newItem.price;

Item\* temp = new Item[size + 1];

for (int i = 0; i < size; ++i) {

temp[i] = inventory[i];

}

temp[size] = newItem;

delete[] inventory;

inventory = temp;

++size;

cout << "Item added successfully!" << endl;

}

void InventoryManager::displayItems() {

cout << "Inventory Items:" << endl;

for (int i = 0; i < size; ++i) {

cout << "Name: " << inventory[i].name << ", Quantity: " << inventory[i].quantity << ", Price: " << inventory[i].price << endl;

}

}

void InventoryManager::deleteItem() {

string name;

cout << "Enter the name of the item to delete: ";

cin >> name;

int index = -1;

for (int i = 0; i < size; ++i) {

if (inventory[i].name == name) {

index = i;

break;

}

}

if (index == -1) {

cout << "Item not found!" << endl;

return;

}

Item\* temp = new Item[size - 1];

for (int i = 0, j = 0; i < size; ++i) {

if (i != index) {

temp[j++] = inventory[i];

}

}

delete[] inventory;

inventory = temp;

--size;

cout << "Item deleted successfully!" << endl;

}

int main() {

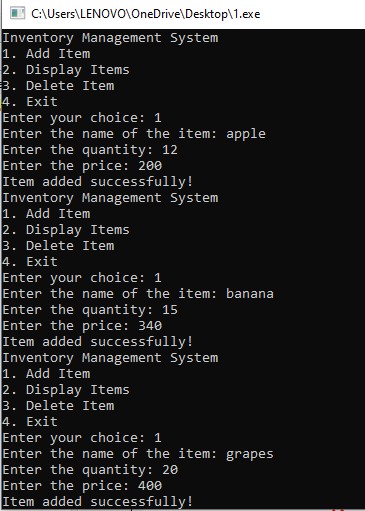
InventoryManager manager;

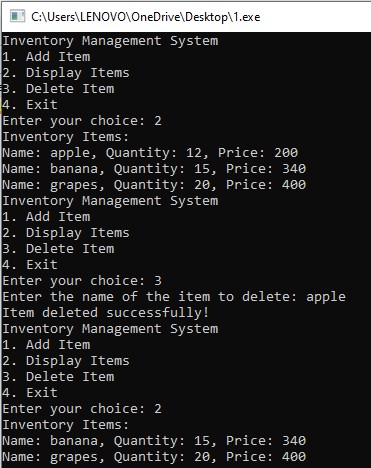
manager.run();

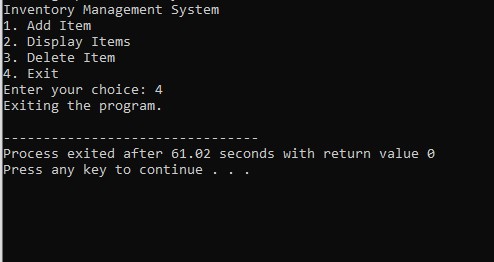
return 0;

}

**Output: -**







**Explanation: -**

1. Include the necessary header files: <iostream> for input-output operations and <string> for using the string data type.
2. Use the standard namespace. It allows you to use classes and functions in the std namespace without qualifying them with std::.
3. Define a class named InventoryManager.
4. Public Section: - Declare a constructor, destructor, and a public method run ().
5. Private Section: - Define a nested private struct named Item with three fields
6. Declare a pointer named inventory for dynamic allocation of an array of Item objects, and an integer size to keep track of the number of items
7. Constructor: - Initialize inventory pointer to nullptr and size to 0.
8. Destructor: - Free the dynamically allocated memory if inventory is not null.
9. Contains the main loop for running the inventory system.
10. Display the menu options.
11. Adds a new item to the inventory.
12. Displays all items in the inventory.
13. Deletes an item from the inventory based on its name.
14. Creates an object of InventoryManager and runs the system.