#include <iostream>

#include <iomanip>

#include <vector>

#include <string>

using namespace std;

// Define a structure to represent a food item

struct FoodItem {

string name;

float price;

};

// Define a class to manage the food ordering system

class FoodOrderingSystem {

private:

vector<FoodItem> menu;

vector<pair<int, vector<int>>> orders; // Pair of order ID and list of item indices

int currentOrderId;

public:

FoodOrderingSystem() : currentOrderId(1) {

// Initialize the menu with some items

menu.push\_back({ "Burger", 5.99 });

menu.push\_back({ "Pizza", 8.99 });

menu.push\_back({ "Pasta", 7.49 });

menu.push\_back({ "Salad", 4.99 });

menu.push\_back({ "Soda", 1.49 });

}

void displayMenu() {

cout << "Menu:\n";

for (size\_t i = 0; i < menu.size(); ++i) {

cout << i + 1 << ". " << menu[i].name << " - $" << fixed << setprecision(2) << menu[i].price << "\n";

}

cout << "0. Exit\n";

}

void takeOrder() {

vector<int> order;

int choice;

cout << "Enter the number of the item you want to order (0 to finish): ";

while (cin >> choice && choice != 0) {

if (choice > 0 && choice <= menu.size()) {

order.push\_back(choice - 1); // Store the index of the chosen item

cout << "Added " << menu[choice - 1].name << " to your order.\n";

}

else {

cout << "Invalid choice, please try again.\n";

}

cout << "Enter the number of the item you want to order (0 to finish): ";

}

if (!order.empty()) {

orders.push\_back({ currentOrderId++, order }); // Assign an ID and store the order

cout << "Order " << currentOrderId - 1 << " has been placed.\n";

}

}

void displayOrder(int orderId) {

bool found = false;

for (const auto& o : orders) {

if (o.first == orderId) {

found = true;

float total = 0.0;

cout << "Order ID " << orderId << ":\n";

for (int index : o.second) {

cout << menu[index].name << " - $" << fixed << setprecision(2) << menu[index].price << "\n";

total += menu[index].price;

}

cout << "Total amount due: $" << fixed << setprecision(2) << total << "\n";

break;

}

}

if (!found) {

cout << "Order ID " << orderId << " not found.\n";

}

}

void displayAllOrders() {

for (const auto& o : orders) {

displayOrder(o.first); // Display each order

}

}

};

int main() {

FoodOrderingSystem system;

int choice;

do {

system.displayMenu();

cout << "Enter your choice: ";

cin >> choice;

if (choice == 0) {

break;

}

switch (choice) {

case 1: case 2: case 3: case 4: case 5:

system.takeOrder();

break;

default:

cout << "Invalid choice, please try again.\n";

break;

}

} while (choice != 0);

cout << "Displaying all orders:\n";

system.displayAllOrders();

cout << "Thank you for your order!\n";

return 0;

}