**SOFTWARE ENGINEERING DEPARTMENT**

**Total Marks: 100**

**Obtained Marks:**

**Project Assignment**

**Last date of Submission: 28 May 2025**

**Submitted To: Shakeel Ahmad**

**Student Name: M.Sohaib Kamran (4784-FOC/BSSE/F23)**

**Soban Abbass (4805-FOC/BSSE/F23)**

**M.Huzaifa**  **(4801-FOC/BSSE/F23)**

**M.Zayd Dawood (4818-FOC/BSSE/F23)**

**Data Structures & Algorithms – Group Project Assignment**

Semester Project | Group Size: 4 Students | Programming Language: C++

**Gym Management System Code:-**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**const int MAX\_MEMBERS = 25;**

**struct Registration {**

**string name, Job\_title, e\_mail, address;**

**long double cnic\_number, cell\_number;**

**};**

**struct Payment {**

**string payment\_method;**

**double amount;**

**int payment\_pin;**

**};**

**struct Membership\_type {**

**string plan;**

**};**

**Registration members[MAX\_MEMBERS];**

**Payment payments[MAX\_MEMBERS];**

**Membership\_type memberships[MAX\_MEMBERS];**

**int num\_members = 0;**

**int search\_member\_by\_cnic(long double cnic) {**

**for (int i = 0; i < num\_members; i++) {**

**if (members[i].cnic\_number == cnic) {**

**return i;**

**}**

**}**

**return -1;**

**}**

**int search\_member\_by\_name(const string& name) {**

**for (int i = 0; i < num\_members; i++) {**

**if (members[i].name == name) {**

**return i;**

**}**

**}**

**return -1;**

**}**

**void sort\_members\_by\_name() {**

**for (int i = 0; i < num\_members - 1; i++) {**

**for (int j = 0; j < num\_members - i - 1; j++) {**

**if (members[j].name > members[j + 1].name) {**

**swap(members[j], members[j + 1]);**

**swap(payments[j], payments[j + 1]);**

**swap(memberships[j], memberships[j + 1]);**

**}**

**}**

**}**

**}**

**void sort\_members\_by\_payment\_amount() {**

**for (int i = 0; i < num\_members - 1; i++) {**

**for (int j = 0; j < num\_members - i - 1; j++) {**

**if (payments[j].amount > payments[j + 1].amount) {**

**swap(members[j], members[j + 1]);**

**swap(payments[j], payments[j + 1]);**

**swap(memberships[j], memberships[j + 1]);**

**}**

**}**

**}**

**}**

**void display\_members(const Registration members[], const Payment payments[], const Membership\_type memberships[], int num\_members) {**

**if (num\_members == 0) {**

**cout << "\nNo members added yet." << endl;**

**}**

**else {**

**cout << "\nMember Details:" << endl;**

**for (int i = 0; i < num\_members; ++i) {**

**cout << "\nMember " << i + 1 << ":" << endl;**

**cout << "Name: " << members[i].name << endl;**

**cout << "Mobile number: " << members[i].cell\_number << endl;**

**cout << "CNIC number: " << members[i].cnic\_number << endl;**

**cout << "E-mail: " << members[i].e\_mail << endl;**

**cout << "Job Title: " << members[i].Job\_title << endl;**

**cout << "Membership Plan: " << memberships[i].plan << endl;**

**if (!payments[i].payment\_method.empty()) {**

**cout << "\nPayment details: " << endl;**

**cout << "Payment method: " << payments[i].payment\_method << endl;**

**cout << "Payment amount: " << payments[i].amount << endl;**

**if (payments[i].payment\_method == "Credit Card" || payments[i].payment\_method == "Visa card") {**

**cout << "Payment PIN: " << payments[i].payment\_pin << endl;**

**}**

**}**

**else {**

**cout << "No payment information available for this member." << endl;**

**}**

**}**

**}**

**}**

**void add\_member() {**

**if (num\_members < MAX\_MEMBERS) {**

**Registration user;**

**Payment payment;**

**Membership\_type membership;**

**cout << "\nEnter member's name: ";**

**cin.ignore();**

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

**cout << "Enter job title: ";**

**getline(cin, user.Job\_title);**

**cout << "Enter email: ";**

**getline(cin, user.e\_mail);**

**cout << "Enter address: ";**

**getline(cin, user.address);**

**cout << "Enter CNIC number: ";**

**cin >> user.cnic\_number;**

**cout << "Enter mobile number: ";**

**cin >> user.cell\_number;**

**cout << "Enter payment method (Credit Card, Paypal, Visa Card): ";**

**cin.ignore();**

**getline(cin, payment.payment\_method);**

**cout << "Enter payment amount: ";**

**cin >> payment.amount;**

**if (payment.payment\_method == "Credit Card" || payment.payment\_method == "Visa Card") {**

**cout << "Enter payment PIN: ";**

**cin >> payment.payment\_pin;**

**}**

**cout << "Enter membership plan (Gold, Platinum, Bronze): ";**

**cin.ignore();**

**getline(cin, membership.plan);**

**members[num\_members] = user;**

**payments[num\_members] = payment;**

**memberships[num\_members] = membership;**

**num\_members++;**

**cout << "Member added successfully!" << endl;**

**} else {**

**cout << "Maximum member limit reached!" << endl;**

**}**

**}**

**int main() {**

**int choice;**

**do {**

**cout << "\n1. Add member\n2. Display all members\n3. Search member by CNIC\n4. Search member by name\n5. Sort members by name\n6. Sort members by payment amount\n7. Exit\n";**

**cout << "Enter your choice: ";**

**cin >> choice;**

**switch (choice) {**

**case 1:**

**add\_member();**

**break;**

**case 2:**

**display\_members(members, payments, memberships, num\_members);**

**break;**

**case 3:**

**{**

**long double cnic;**

**cout << "Enter CNIC to search: ";**

**cin >> cnic;**

**int index = search\_member\_by\_cnic(cnic);**

**if (index != -1) {**

**cout << "Member found: " << members[index].name << endl;**

**} else {**

**cout << "Member not found!" << endl;**

**}**

**}**

**break;**

**case 4:**

**{**

**string name;**

**cout << "Enter name to search: ";**

**cin.ignore();**

**getline(cin, name);**

**int index = search\_member\_by\_name(name);**

**if (index != -1) {**

**cout << "Member found: " << members[index].name << endl;**

**} else {**

**cout << "Member not found!" << endl;**

**}**

**}**

**break;**

**case 5:**

**sort\_members\_by\_name();**

**cout << "Members sorted by name." << endl;**

**break;**

**case 6:**

**sort\_members\_by\_payment\_amount();**

**cout << "Members sorted by payment amount." << endl;**

**break;**

**case 7:**

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

**break;**

**default:**

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

**break;**

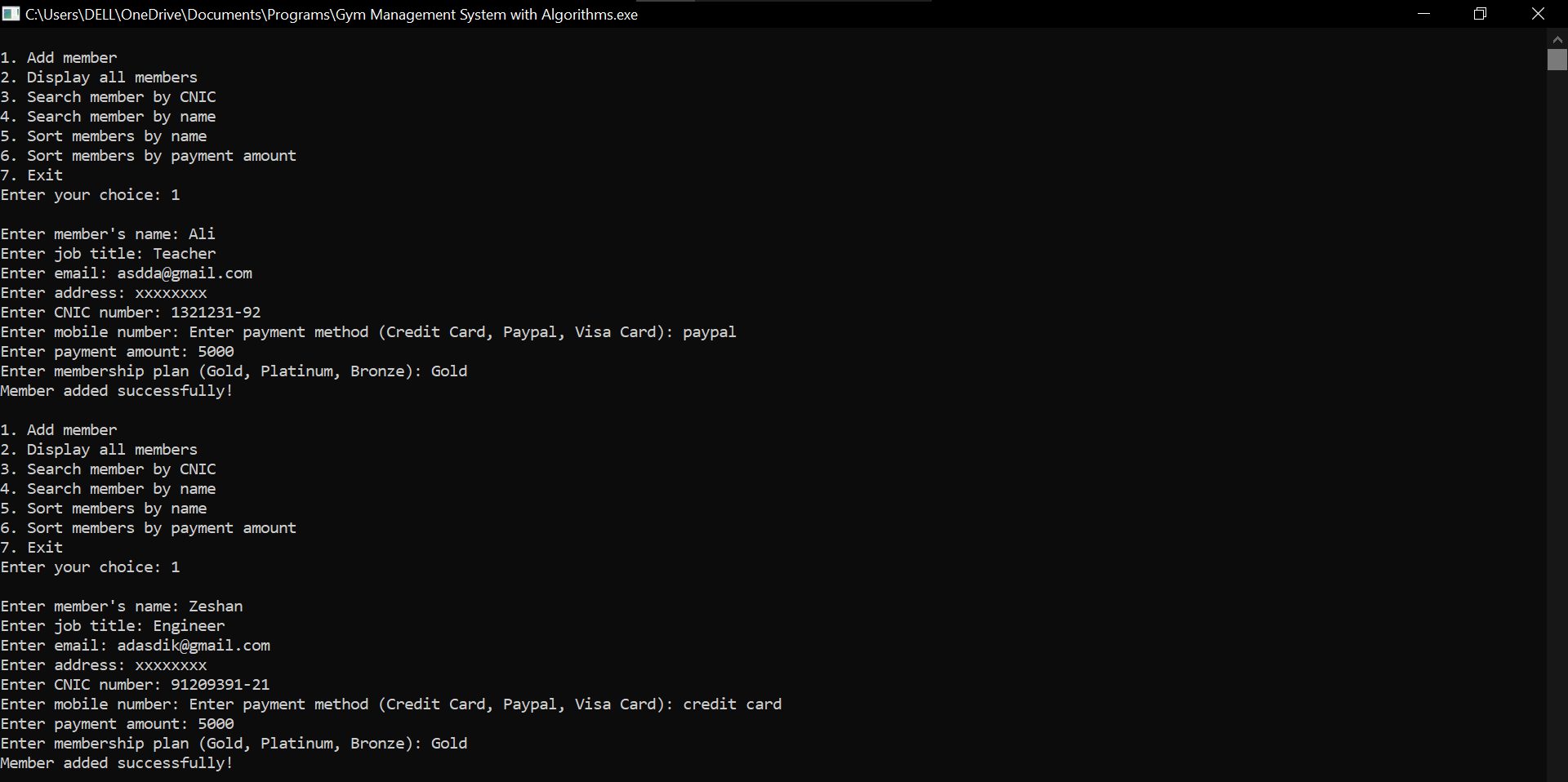
**}**

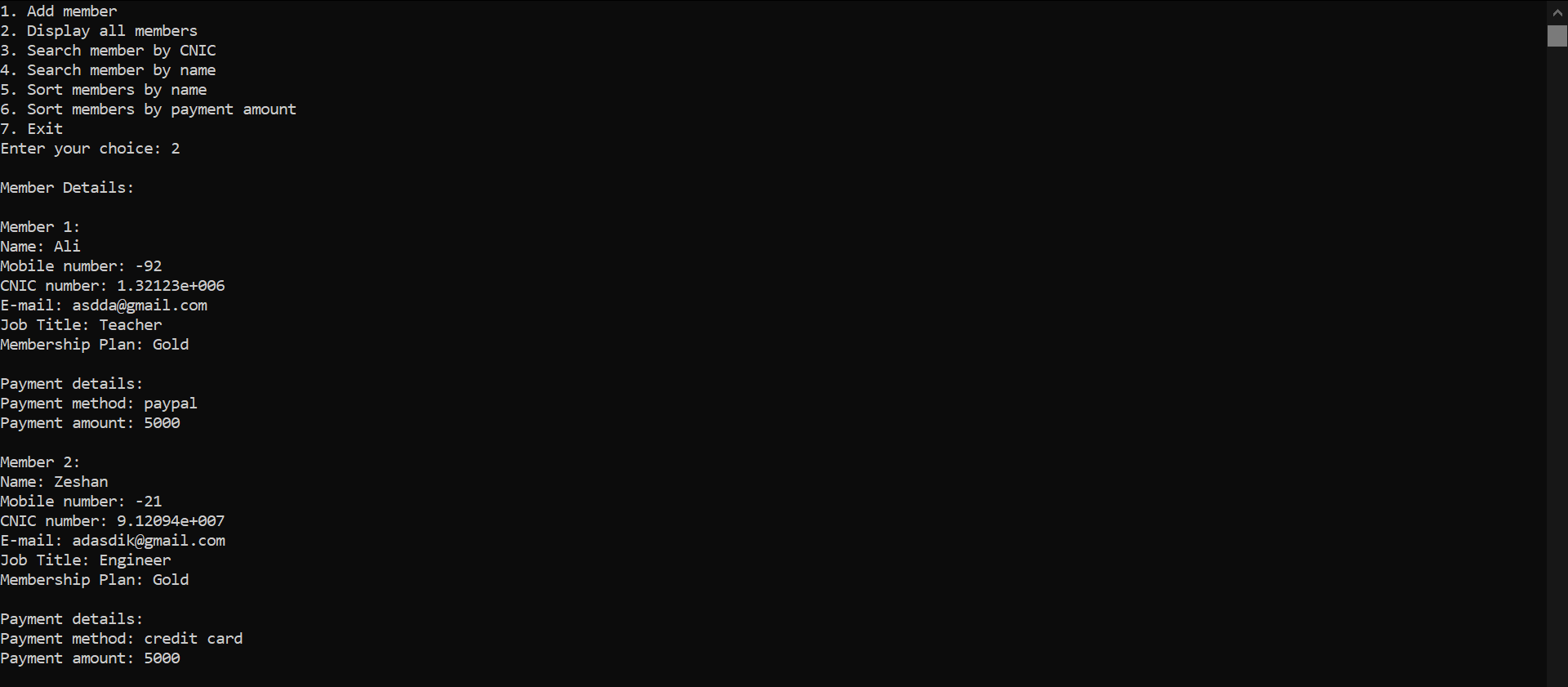
**} while (choice != 7);**

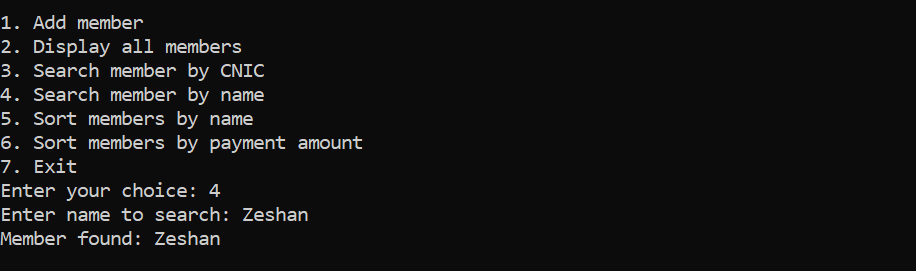
**return 0;**

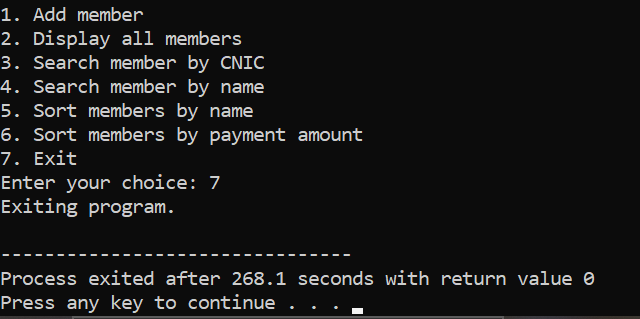
**}**

**OUTPUT:-**

****

****

****

****

**GitHub Link:**

[**zayddawood-0/DSA-Project: The topic of my project is Gym Management system.**](https://github.com/zayddawood-0/DSA-Project)

**Linkedln Post links:**

* [**https://www.linkedin.com/posts/m-sohaib-kamran-5b6529369\_gym-management-system-code-activity-7336115084546498562-4JHr?utm\_source=share&utm\_medium=member\_android&rcm=ACoAAFthbhoB\_T32i0DU-rX-rBfWf8FYKfvOAVI**](https://www.linkedin.com/posts/m-sohaib-kamran-5b6529369_gym-management-system-code-activity-7336115084546498562-4JHr?utm_source=share&utm_medium=member_android&rcm=ACoAAFthbhoB_T32i0DU-rX-rBfWf8FYKfvOAVI)
* [**https://www.linkedin.com/posts/muhammad-huzaifa-88a831323\_dsa-project-gym-management-system-activity-7336115288876232707-Pr-e?utm\_source=share&utm\_medium=member\_desktop&rcm=ACoAAFHQjqoB2WGIknX7cJwXkXWy41oqhFmhkqg**](https://www.linkedin.com/posts/muhammad-huzaifa-88a831323_dsa-project-gym-management-system-activity-7336115288876232707-Pr-e?utm_source=share&utm_medium=member_desktop&rcm=ACoAAFHQjqoB2WGIknX7cJwXkXWy41oqhFmhkqg)
* [**https://www.linkedin.com/posts/m-zayd-dawood-705764352\_gym-management-system-using-c-activity-7336121271635648518-O\_Lz?utm\_source=share&utm\_medium=member\_android&rcm=ACoAAFgIePUBdDVERsE9WWTYJQwJ34wdjZ55ibo**](https://www.linkedin.com/posts/m-zayd-dawood-705764352_gym-management-system-using-c-activity-7336121271635648518-O_Lz?utm_source=share&utm_medium=member_android&rcm=ACoAAFgIePUBdDVERsE9WWTYJQwJ34wdjZ55ibo)