

OOP-LAB Task 7

Roll No: 24P-0706

Dept: BS-CS

Name: Aazan Noor Khuwaja

Section : 2D

QNS-1:

```
#include<iostream>
```

```
using namespace std;
```

```
class room {
```

```
    int room_no;
```

```
public:
```

```
    room(int r = 0) : room_no(r) {} // Default value needed
```

```
    int show_room() {
```

```
        return room_no;
```

```
    }
```

```
};
```

```
class doctor {
```

```
    int id;
```

```
    string name;
```

```
    string specialization; // spelling fixed
```

```
    room r1;
```

```
public:
```

```
    doctor() : id(0), name(""), specialization(""), r1(0) {} // Default constructor needed
```

```
    doctor(int i, string n, string spec, int ro) : id(i), name(n), specialization(spec), r1(ro) {}
```

```
    string get_name() {
```

```

        return name;
    }

    string get_specialization() {
        return specialization;
    }

    friend void doctor_details(doctor doc);
};

```

```

void doctor_details(doctor doc) {
    cout << "Doctor Id: " << doc.id << endl;
    cout << "Doctor Name: " << doc.name << endl;
    cout << "Doctor Specialization: " << doc.specialization << endl;
    cout << "Doctor Room No: " << doc.r1.show_room() << endl;
}

```

```

class patient {
    int pat_id;
    string pat_name;
    string pat_disease;
    doctor* d1;
public:
    patient() : pat_id(0), pat_name(""), pat_disease(""), d1(nullptr) {}
    patient(int i, string n, string d, doctor* doct) : pat_id(i), pat_name(n), pat_disease(d), d1(doct) {}

    friend void show_patient(patient p1);
};

```

```

void show_patient(patient p1) {
    cout << "Patient Id: " << p1.pat_id << endl;
}

```

```
    cout << "Patient Name: " << p1.pat_name << endl;
    cout << "Patient Disease: " << p1.pat_disease << endl;
    cout << "Doctor Name: " << p1.d1->get_name() << endl;
    cout << "Doctor Specialization: " << p1.d1->get_specialization() << endl;
    cout << endl;
}
```

```
class hospital {
public:
    patient p1[10];
    doctor d1[10];
    int doc_count;
    int pat_count;

    hospital() : doc_count(0), pat_count(0) {}

    void printHospitalInfo() {
        cout << "Hospital Patients Information:" << endl;
        for (int i = 0; i < pat_count; i++) {
            show_patient(p1[i]);
        }
    }

    void show_doctor_details() {
        cout << "Hospital Doctors Information" << endl;
        for (int i = 0; i < doc_count; i++) {
            doctor_details(d1[i]);
        }
    }
}
```

```
};
```

```
int main() {  
    hospital h;  
    h.d1[0] = doctor(1, "Dr. Ghulam Nabi", "MBBS GYNO", 101);  
    h.d1[1] = doctor(2, "Dr. Qasim Jan", "Neurologist", 102);  
    h.doc_count = 2;  
    h.p1[0] = patient(1, "Azan", "Heart Disease", &h.d1[0]);  
    h.p1[1] = patient(2, "Aslam", "Brain Tumor", &h.d1[1]);  
    h.pat_count = 2;  
    h.show_doctor_details();  
    cout << endl;  
    h.printHospitalInfo();  
  
    return 0;  
}
```

Output:

```
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-7> ./output
Hospital Doctors Information
Doctor Id: 1
Doctor Name: Dr. Ghulam Nabi
Doctor Specialization: MBBS GYNO
Doctor Room No: 101
Doctor Id: 2
Doctor Name: Dr. Qasim Jan
Doctor Specialization: Neurologist
Doctor Room No: 102

Hospital Patients Information:
Patient Id: 1
Patient Name: Azan
Patient Disease: Heart Disease
Doctor Name: Dr. Ghulam Nabi
Doctor Specialization: MBBS GYNO

Patient Id: 2
Patient Name: Aslam
Patient Disease: Brain Tumor
Doctor Name: Dr. Qasim Jan
Doctor Specialization: Neurologist

PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-7> 
```

QNS-2:

Code:

```
#include<iostream>

#include<string>

using namespace std;

class Professor {

    string professor_name;

    int employee_id;
```

public:

Professor(string name, int id) : professor_name(name), employee_id(id) {}

void display_professor() {

cout << "Professor Name: " << professor_name << ", Employee ID: " << employee_id << endl;

}

};

class Student;

class Course {

string course_title;

string course_code;

Professor* prof;

Student* students[30];

int student_count;

string hidden_notes;

float course_stars;

public:

**Course(string title, string code, Professor* p) : course_title(title), course_code(code), prof(p),
 student_count(0) {**

hidden_notes = "Internal: Focus on research papers.";

course_stars = 4.6;

}

void assign_professor(Professor* p) {

prof = p;

}

```
void enroll_student(Student* s) {  
    if (student_count < 30) {  
        students[student_count++] = s;  
    }  
}
```

```
void show_course() {  
    cout << "Course Title: " << course_title << ", Course Code: " << course_code << endl;  
    if (prof != nullptr) {  
        prof->display_professor();  
    } else {  
        cout << "No Professor assigned yet.\n";  
    }  
    cout << "Total Students Enrolled: " << student_count << endl;  
}
```

```
friend class Student;  
};
```

```
class Student {  
    string student_name;  
    int roll_number;
```

```
public:
```

```
    Student(string name, int roll) : student_name(name), roll_number(roll) {}
```

```
void view_course_details(Course &c) {  
    cout << "\nStudent View" << endl;
```

```
        cout << "Course Title: " << c.course_title << ", Course Code: " << c.course_code << endl;
        cout << "Internal Notes: " << c.hidden_notes << endl;
        cout << "Course Rating: " << c.course_stars << " stars" << endl;
    }
};
```

```
int main() {
    Professor prof1("Dr. Sarah", 501);
    Professor prof2("Dr. Malik", 502);

    Course course1("Data Structures", "CS201", &prof1);
    Course course2("Database Systems", "CS301", &prof2);

    Student student1("Alice", 1001);
    Student student2("Bob", 1002);

    course1.enroll_student(&student1);
    course1.enroll_student(&student2);

    course2.enroll_student(&student1);

    cout << "Course 1 Details:" << endl;
    course1.show_course();

    cout << "\nCourse 2 Details:" << endl;
    course2.show_course();

    cout << "\nStudent Accessing Course Details:" << endl;
    student1.view_course_details(course1);
    student2.view_course_details(course2);
```

```
    return 0;
}
```

Output:

```
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-7> ./output
Course 1 Details:
Course Title: Data Structures, Course Code: CS201
Professor Name: Dr. Sarah, Employee ID: 501
Total Students Enrolled: 2

Course 2 Details:
Course Title: Database Systems, Course Code: CS301
Professor Name: Dr. Malik, Employee ID: 502
Total Students Enrolled: 1

Student Accessing Course Details:

Student View
Course Title: Data Structures, Course Code: CS201
Internal Notes: Internal: Focus on research papers.
Course Rating: 4.6 stars

Student View
Course Title: Database Systems, Course Code: CS301
Internal Notes: Internal: Focus on research papers.
Course Rating: 4.6 stars
PS C:\Users\Azan Noor\OneDrive\Desktop\Lab Task Opp\labtask-7> █
```