Open-Ended Problem (OEP)

- **Enrollment no : IU2341231570(Anurag Barkhade)**
- **Sub : Data Structure and Algorithms**
- **Sub-Code : CE0417**
- **Name of Faculty : Ms.Zalak Vyas**

Open-Ended Problem Statement: "Student Information Management System" (SIMS).

○ Following are Solution Approaches :

Vectors (std::vector):
Vectors are used to store collections of students, courses, and attendance records. They provide dynamic array-like functionality, allowing for easy addition and removal of elements.
Maps (std::map):
Maps are used to associate course IDs with lists of enrolled student IDs. They provide an efficient way to store key-value pairs, allowing for quick retrieval of information based on a specific key.
Structures:
Structures are used to define the Student, Course, and Attendance data types. These structures group related data together, making it easier to manage and manipulate student information, course details, and attendance records.

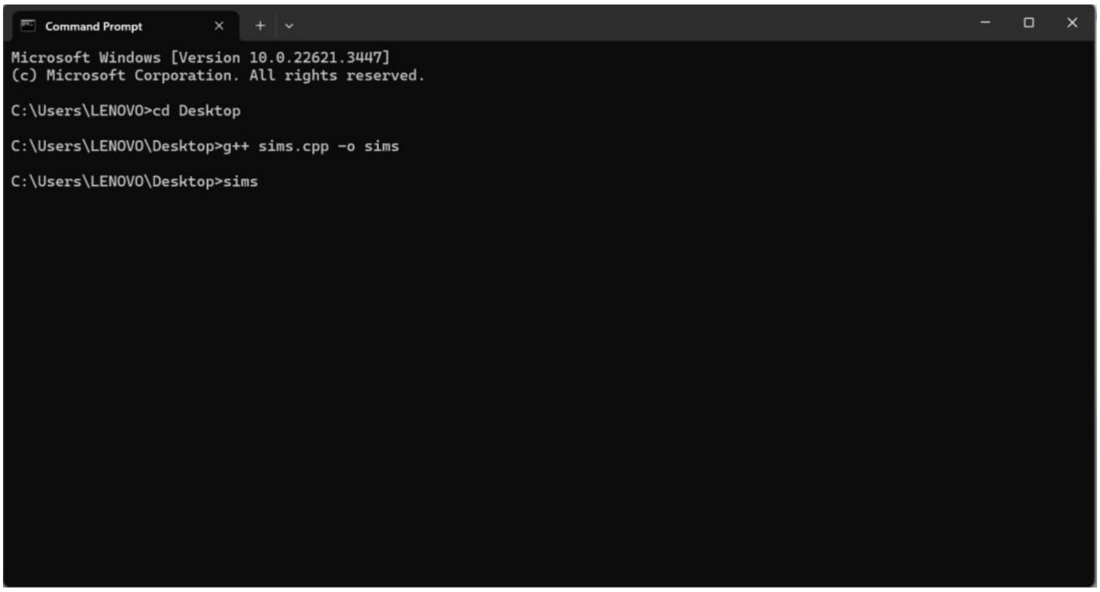
○ Code :

```
#include <iostream>
#include <vector>
#include <map>

using namespace std;
```

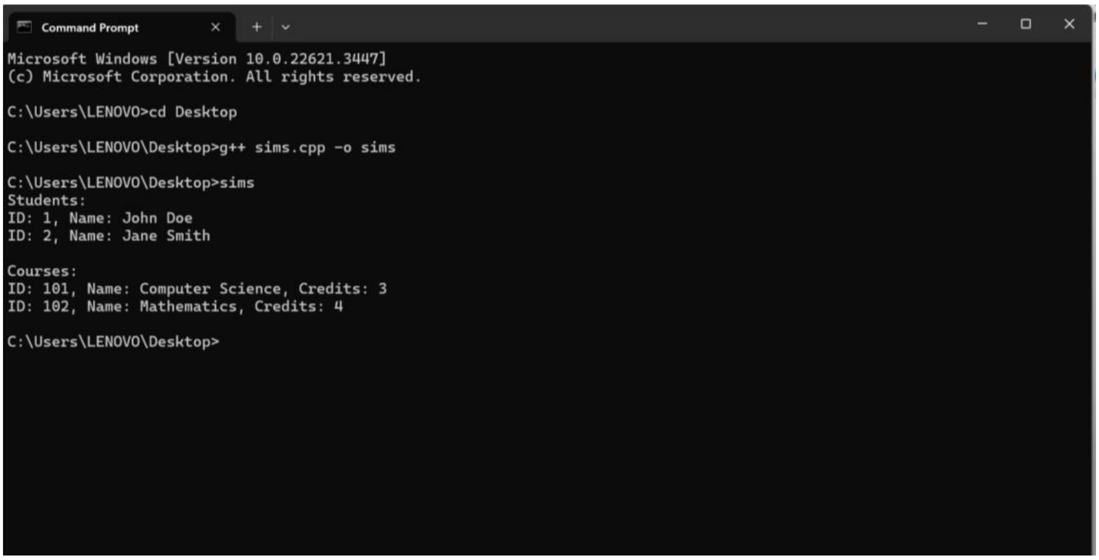
```
// Define structures for Student, Course, and Attendance
struct Student {     int id;     string name;
    string address;
    string contact;
};

struct Course {
    int id;
string name;
int credits;
};

struct Attendance {
int studentId;     int
courseId;
    int attendancePercentage;
};

// Define SIMS class to manage students, courses, and attendance
class SIMS { private:
    vector<Student> students;
vector<Course> courses;
vector<Attendance> attendanceRecords;
    map<int, vector<int>> courseEnrollment;

public:
    // Functions to add, remove, and display students
void addStudent(const Student& student) {
        students.push_back(student);
    }

    void removeStudent(int studentId) {
        // Implement removal logic
    }

    void displayStudents() const {         for
(const auto& student : students) {
            cout << "ID: " << student.id << ", Name: " << student.name << endl;
        }
    }

    // Functions to add, remove, and display courses
void addCourse(const Course& course) {
courses.push_back(course);
    }

    void removeCourse(int courseId) {
        // Implement removal logic
    }
```

```
   void displayCourses() const {         for
(const auto& course : courses) {
cout << "ID: " << course.id << ",
Name: " << course.name << ", Credits:
" << course.credits << endl;
      }
   }

   // Function to enroll students in courses
   void enrollStudentInCourse(int studentId, int courseId) {
courseEnrollment[courseId].push_back(studentId);
   }

   // Function to record student attendance
   void recordAttendance(int studentId, int courseId, int attendancePercentage) {
attendanceRecords.push_back({studentId, courseId, attendancePercentage});
   }
};

int main() {
   // Create SIMS object
   SIMS sims;

   // Sample data
   Student student1 = {1, "John Doe", "123 Main St", "123-456-7890"};
Student student2 = {2, "Jane Smith", "456 Elm St", "987-654-3210"};
sims.addStudent(student1);     sims.addStudent(student2);

   Course course1 = {101, "Computer Science", 3};
Course course2 = {102, "Mathematics", 4};
sims.addCourse(course1);     sims.addCourse(course2);

   sims.enrollStudentInCourse(1, 101);
sims.enrollStudentInCourse(2, 101);
sims.enrollStudentInCourse(2, 102);

   sims.recordAttendance(1, 101, 90);
sims.recordAttendance(2, 101, 85);
sims.recordAttendance(2, 102, 95);

   // Display students and courses
cout << "Students:" << endl;
   sims.displayStudents();

   cout << "\nCourses:" << endl;
sims.displayCourses();

   return 0;
}
```

- **Output :**



```
Microsoft Windows [Version 10.0.22621.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>cd Desktop

C:\Users\LENOVO\Desktop>g++ sims.cpp -o sims

C:\Users\LENOVO\Desktop>sims
```



```
Microsoft Windows [Version 10.0.22621.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>cd Desktop

C:\Users\LENOVO\Desktop>g++ sims.cpp -o sims

C:\Users\LENOVO\Desktop>sims
Students:
ID: 1, Name: John Doe
ID: 2, Name: Jane Smith

Courses:
ID: 101, Name: Computer Science, Credits: 3
ID: 102, Name: Mathematics, Credits: 4

C:\Users\LENOVO\Desktop>
```

- **Conclusion :** Thus,In this way we have explained the open ended problem statement of **Student Information Management System** and solve it.