Open-Ended Problem (OEP)

- **Enrollment no :IU2341231571(Anurag Barkhade)**
- **Sub : Data Structure and Algorithms**
- **Sub-Code : CE0417**
- **Name of Faculty : Ms.Zalak Vyas**

Open-Ended Problem Statement: **Student Gradebook**

You are tasked with creating a student gradebook system for a school. The system should allow teachers to enter grades for each student in different subjects and calculate overall grades and averages. It should also provide functionalities for adding new students, adding grades, and generating reports.

Solution:

Data Structures:

Map: To store student information where the key is the student's ID and the value is a struct containing student details and grades.

Vector/Array: To store grades for each subject.

Functions:

Add Student:
Allows teachers to add a new student to the gradebook system along with their details.

Add Grades:
Enables teachers to add grades for each subject for a specific student.

Calculate Average:
Calculates the average grade for each student across all subjects.

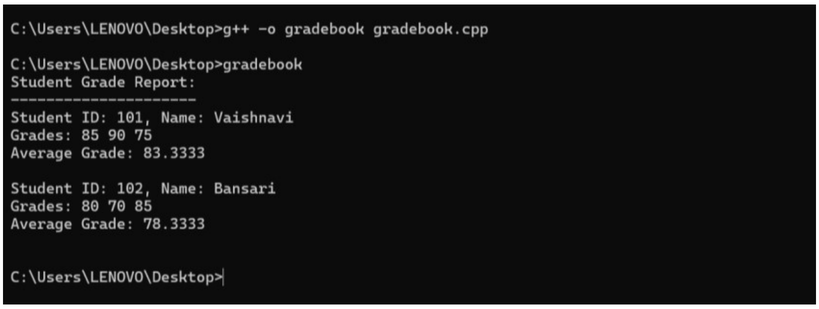
Generate Report:
Generates a report listing all students along with their grades and averages.

Code :

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

using namespace std;

struct Student {
string name;
    vector<int> grades;

    // Constructor with default name

  Student(const string& n = "Unknown") : name(n) {}
};

class Gradebook { private:
    map<int, Student> students;

public:

void addStudent(int id, const string& name) {
        students[id] = Student(name);
    }

    void addGrade(int id, int grade) {
        students[id].grades.push_back(grade);
    }

    float calculateAverage(int id) {

const vector<int>& grades = students[id].grades;
float sum = 0;         for (int grade : grades) {
            sum += grade;
        }
        return sum / grades.size();
    }

    void generateReport() {
```

```
 cout << "Student Grade Report:" << endl;
cout << "----------------------" << endl;        for
(const auto& pair : students) {
            int id = pair.first;
            const Student& student = pair.second;
            cout << "Student ID: " << id << ", Name: " << student.name << endl;
cout << "Grades: ";            for (int grade : student.grades) {
                cout << grade << " ";
            }
            cout << endl;
            cout << "Average Grade: " << calculateAverage(id) << endl;
cout << endl;
        }
    }
};


int main() {
    Gradebook gradebook;

    gradebook.addStudent(101, "Vaishnavi");
gradebook.addGrade(101, 85);     gradebook.addGrade(101,
90);     gradebook.addGrade(101, 75);

    gradebook.addStudent(102, "Bansari");
gradebook.addGrade(102, 80);     gradebook.addGrade(102,
70);     gradebook.addGrade(102, 85);

    gradebook.generateReport();

    return 0;
}
```

Output :



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

C:\Users\LENOVO\Desktop>gradebook
Student Grade Report:
---------------------
Student ID: 101, Name: Vaishnavi
Grades: 85 90 75
Average Grade: 83.3333

Student ID: 102, Name: Bansari
Grades: 80 70 85
Average Grade: 78.3333


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

Conclusion : Thus,In this way we have explained the open ended problem statement of Student Gradebook and solve it.