**Name of Student: Raheel Kotwal**

**Roll Number: 45**

**Experiment Number: 17**

**Title: Student Management**

**Theory:**  The code defines a Student class and a StudentRecord class to manage student details, allowing the user to input information for a specified number of students, displaying their details and calculating the average marks. The program uses a random roll number generator and limits the number of students to 100

**Code:**

**#include <iostream>**

**#include <cstdlib>**

**#include <ctime>**

**using namespace std;**

**class Student {**

**private:**

**string name;**

**int rollNumber;**

**float marks;**

**public:**

**Student()**

**: name(""), rollNumber(0), marks(0.0) {}**

**Student(const string& studentName, int studentRollNumber, float studentMarks)**

**: name(studentName), rollNumber(studentRollNumber), marks(studentMarks) {}**

**void displayDetails() const {**

**cout << "Name: " << name << "\n"<< "Roll Number: " << rollNumber << "\n"<< "Marks: " << marks << "\n\n";**

**}**

**float getMarks() const {**

**return marks;**

**}**

**int getRollNumber() const {**

**return rollNumber;**

**}**

**};**

**class StudentRecord {**

**private:**

**static const int MAX\_STUDENTS = 100; //Not more than 100 student details can be stored in this code (code limitation)**

**Student students[MAX\_STUDENTS];**

**int numStudents;**

**public:**

**StudentRecord() : numStudents(0) {}**

**void addStudent(const Student& student) {**

**if (numStudents < MAX\_STUDENTS) {**

**students[numStudents] = student;**

**numStudents++;**

**} else {**

**cout << "Maximum number of students reached.\n";**

**}**

**}**

**void displayAllStudents() const {**

**for (int i = 0; i < numStudents; ++i) {**

**students[i].displayDetails();**

**}**

**}**

**void calculateAndDisplayAverageMarks() const {**

**if (numStudents == 0) {**

**cout << "No students in the record.\n";**

**return;**

**}**

**float totalMarks = 0.0;**

**for (int i = 0; i < numStudents; ++i) {**

**totalMarks += students[i].getMarks();**

**}**

**float averageMarks = totalMarks / numStudents;**

**cout << "Average Marks of all students: " << averageMarks << "\n";**

**}**

**int generateRandomRollNumber() {**

**int newRollNumber;**

**bool isUnique;**

**do {**

**newRollNumber = rand() % 1000 + 1;**

**isUnique = true;**

**for (int i = 0; i < numStudents; ++i) {**

**if (students[i].getRollNumber() == newRollNumber) {**

**isUnique = false;**

**break;**

**}**

**}**

**} while (!isUnique);**

**return newRollNumber;**

**}**

**};**

**int main() {**

**srand(static\_cast<unsigned int>(time(0)));**

**StudentRecord studentRecord;**

**int numStudents;**

**cout << "Enter the number of students: ";**

**cin >> numStudents;**

**for (int i = 0; i < numStudents; ++i) {**

**cin.ignore();**

**string name;**

**float marks;**

**cout << "\nEnter details for student " << i + 1 << ":\n";**

**cout << "Name: ";**

**getline(cin, name);**

**cout << "Marks: ";**

**cin >> marks;**

**int rollNumber = studentRecord.generateRandomRollNumber();**

**Student newStudent(name, rollNumber, marks);**

**studentRecord.addStudent(newStudent);**

**}**

**cout << "\nStudent Details:\n";**

**studentRecord.displayAllStudents();**

**studentRecord.calculateAndDisplayAverageMarks();**

**return 0;**

**}**

**Output:(screenshot):**



