Name-Suryaprakash Yadav

Class – SE B

Batch- S2

****

AUTOMATED TERM WORK ASSESMENT

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <iomanip>
using namespace std;

class Student {
public:
   int student_id;
   string name;
   double attendance_score;
   double unit_test_score;
   double prelim_score;
   double achievements_score;
   double mock_practical_score;

   // Constructor to initialize student data
   Student(int id, string student_name) {
      student_id = id;
      name = student_name;
      attendance_score = 0;
      unit_test_score = 0;
      prelim_score = 0;
```

```
    achievements_score = 0;
    mock_practical_score = 0;
}

// Function to calculate final score
double calculate_final_score() {
    double final_score = (attendance_score * 0.20) +
                         (unit_test_score * 0.30) +
                         (achievements_score * 0.10) +
                         (mock_practical_score * 0.40);
    return final_score;
}

// Function to display student details and scores
void display_student_report() {
    cout << "Student ID: " << student_id << endl;
    cout << "Name: " << name << endl;
    cout << "Attendance Score: " << attendance_score << endl;
    cout << "Unit Test Score: " << unit_test_score << endl;
    cout << "Prelim Score: " << prelim_score << endl;
    cout << "Achievements Score: " << achievements_score << endl;
    cout << "Mock Practical Score: " << mock_practical_score << endl;
    cout << "Final Score: "<< calculate_final_score() << endl;
    cout << "Grade: " << get_grade() << endl;
}

// Function to assign grade based on final score
string get_grade() {
    double final_score = calculate_final_score();
    if (final_score >= 90) {
```

```
        return "A";
    } else if (final_score >= 80) {
        return "B";
    } else if (final_score >= 70) {
        return "C";
    } else if (final_score >= 60) {
        return "D";
    } else {
        return "F";
    }
}

// Function to input data for a student
void input_data() {
    int days_attended;
    int working_days;
    cout<<"total no. of working days : ";
    cin>>working_days;
    cout<<"no. of days of attending college : ";
    cin>>days_attended;
    attendance_score=days_attended *100/working_days;
    cout << "Enter Unit Test Score (0-100): ";
    cin >> unit_test_score;
    cout << "Enter Prelim Score (0-100): ";
    cin >> prelim_score;
    cout << "Enter Achievements Score (0-100): ";
    cin >> achievements_score;
    cout << "Enter Mock Practical Score (0-100): ";
    cin >> mock_practical_score;
}
```

```
};

class StudentManagementSystem {
private:
    vector<Student> students;

public:
    void add_student() {
        int id;
        string name;
        cout << "Enter student ID: ";
        cin >> id;
        cout << "Enter student name: ";
        cin.ignore();//ignore characters in the input buffer
        getline(cin, name);
        Student new_student(id, name);
        new_student.input_data();
        students.push_back(new_student);
    }

    // Function to display all students and their reports
    void display_all_students() {
        if (students.empty()) {
            cout << "No students available!" << endl;
            return;
        }
        for (Student &student : students) {
            student.display_student_report();
            cout << "----------------------------------------" << endl;
        }
```

```
}

// Function to save student data to a file
void save_data() {
    ofstream file("students_data.txt");
    if (file.is_open()) {
        for (Student &student : students) {
            file << student.student_id << "," << student.name << ","
                 << student.attendance_score << ","
                 << student.unit_test_score << ","
                 << student.prelim_score << ","
                 << student.achievements_score << ","
                 << student.mock_practical_score << endl;
        }
        file.close();
        cout << "Data saved successfully!" << endl;
    } else {
        cout << "Unable to save data!" << endl;
    }
}

// Function to load student data from a file
void load_data() {
    ifstream file("students_data.txt");
    if (file.is_open()) {
        int id;
        string name;
        double attendance, unit_test, prelim, achievements, mock_practical;
        string line;
        while (getline(file, line)) {
```

```
        size_t pos = 0;
        vector<string> student_data;
        while ((pos = line.find(',')) != string::npos) {
            student_data.push_back(line.substr(0, pos));
            line.erase(0, pos + 1);
        }
        student_data.push_back(line);

        id = stoi(student_data[0]);
        name = student_data[1];
        attendance = stod(student_data[2]);
        unit_test = stod(student_data[3]);
        prelim = stod(student_data[4]);
        achievements = stod(student_data[5]);
        mock_practical = stod(student_data[6]);

        Student new_student(id, name);
        new_student.attendance_score = attendance;
        new_student.unit_test_score = unit_test;
        new_student.prelim_score = prelim;
        new_student.achievements_score = achievements;
        new_student.mock_practical_score = mock_practical;

        students.push_back(new_student);
    }
    file.close();
    cout << "Data loaded successfully!" << endl;
} else {
    cout << "Unable to load data!" << endl;
}
```

```
    }
};

int main() {
    StudentManagementSystem sms;
    int choice;

    sms.load_data(); // Load data when the program starts

    do {
        cout << "\n----- Automated Term Work Assessment System -----\n";
        cout << "1. Add Student\n";
        cout << "2. Display All Students\n";
        cout << "3. Save Data\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                sms.add_student();
                break;
            case 2:
                sms.display_all_students();
                break;
            case 3:
                sms.save_data();
                break;
            case 4:
                cout << "Exiting program..." << endl;
```

```
            break;
        default:
            cout << "Invalid choice! Try again." << endl;
    }
  } while (choice != 4);

  return 0;
}**********************************Output****************************************
----- Automated Term Work Assessment System -----
1. Add Student
2. Display All Students
3. Save Data
4. Exit
Enter your choice: 1
Enter student ID: 278
Enter student name: Chirag Shah
total no. of working days : 200
no. of days of attending college : 156
Enter Unit Test Score (0-100): 45
Enter Prelim Score (0-100): 78
Enter Achievements Score (0-100): 79
Enter Mock Practical Score (0-100): 75

----- Automated Term Work Assessment System -----
1. Add Student
2. Display All Students
3. Save Data
4. Exit
```

Enter your choice: 3

Data saved successfully!

----- Automated Term Work Assessment System -----

1. Add Student

2. Display All Students

3. Save Data

4. Exit

Enter your choice: 2

Student ID: 278

Name: Chirag Shah

Attendance Score: 78

Unit Test Score: 45

Prelim Score: 78

Achievements Score: 79

Mock Practical Score: 75

Final Score: 67

Grade: D

----- Automated Term Work Assessment System -----

1. Add Student

2. Display All Students

3. Save Data

4. Exit

Enter your choice: 4

Exiting program...