NAME-POIMANTI HAZRA

REGISTRATION NUMBER- 22BEC1077

***TITLE-"Attendance Analyzer: Optimizing Class Attendance with Custom Input"***

**Introduction:**

The "Attendance Analyzer: Optimizing Class Attendance with Custom Input" is a program designed to help students optimize their class attendance by analyzing their attendance records. It allows users to input their course timetable along with the total number of classes for each course and the number of classes attended. The program then calculates the attendance percentage for each course and determines the number of classes that can be missed while still meeting a 75% attendance requirement. This tool provides students with valuable insights into their attendance status for each course, helping them manage their time

**Code:**

#include <iostream>

#include <map>

#include <string>

// Function to calculate attendance and number of classes you can miss

void calculateAttendance(std::map<std::string, int>& timetable, std::map<std::string, int>& attendance) {

std::cout << "Attendance Status:" << std::endl;

for (const auto& entry : timetable) {

const std::string& course = entry.first;

int totalClasses = timetable[course];

std::cout << "Course: " << course << std::endl;

std::cout << "Total classes: " << totalClasses << std::endl;

if (attendance.find(course) != attendance.end()) {

int attendedClasses = attendance[course];

double attendancePercentage = (double(attendedClasses) / totalClasses) \* 100.0;

std::cout << "Attended classes: " << attendedClasses << std::endl;

std::cout << "Attendance percentage: " << attendancePercentage << "%" << std::endl;

// Calculate number of classes you can miss (considering 75% attendance required)

int classesRequired = totalClasses \* 0.75;

int classesCanMiss = classesRequired - attendedClasses;

std::cout << "Classes you can miss: " << classesCanMiss << std::endl;

} else {

std::cout << "Attended classes: 0" << std::endl;

std::cout << "Attendance percentage: 0%" << std::endl;

std::cout << "Classes you can miss: " << totalClasses << std::endl;

}

std::cout << std::endl;

}

}

int main() {

// Define timetable (course -> total classes)

std::map<std::string, int> timetable;

// Input total classes for each course

std::cout << "Enter your timetable (course -> total classes):" << std::endl;

for (int i = 1; ; ++i) {

std::string course;

int totalClasses;

std::cout << "Enter course " << i << " name (or 'done' to finish): ";

std::cin >> course;

if (course == "done") break;

std::cout << "Enter total classes for " << course << ": ";

std::cin >> totalClasses;

timetable[course] = totalClasses;

}

// Define attendance (course -> attended classes)

std::map<std::string, int> attendance;

std::cout << "Enter your attendance record (course -> attended classes):" << std::endl;

for (const auto& entry : timetable) {

const std::string& course = entry.first;

int attendedClasses;

std::cout << "Enter attended classes for " << course << ": ";

std::cin >> attendedClasses;

attendance[course] = attendedClasses;

}

// Calculate attendance and number of classes you can miss

calculateAttendance(timetable, attendance);

return 0;

}

