**Code:**

#include <iostream>

#include <vector>

#include <algorithm>

#include "rapidcsv-master/src/rapidcsv.h" // Replace with the correct path

struct StudentResult {

    int ObjectOrientedProgramming;

    int CalculusAndDifferentialEquations;

    int LinearAlgebra;

    int CommunicationSkills;

    int IslamicStudies;

    int EngineeringDrawing;

};

void customSort(std::vector<int>& subjectMarks, int low, int high) {

    if (low < high) {

        int pivot = subjectMarks[high];

        int i = (low - 1);

        for (int j = low; j <= high - 1; ++j) {

            if (subjectMarks[j] < pivot) {

                ++i;

                std::swap(subjectMarks[i], subjectMarks[j]);

            }

        }

        std::swap(subjectMarks[i + 1], subjectMarks[high]);

        int pivotIdx = i + 1;

        customSort(subjectMarks, low, pivotIdx - 1);

        customSort(subjectMarks, pivotIdx + 1, high);

    }

}

std::vector<StudentResult> mergeSubjectMarks(const std::vector<int>& oop,

                                            const std::vector<int>& cald,

                                            const std::vector<int>& ed,

                                            const std::vector<int>& la,

                                            const std::vector<int>& comm,

                                            const std::vector<int>& isl) {

    std::vector<StudentResult> mergedResults;

    size\_t totalStudents = oop.size(); // Assuming all subject vectors have the same size

    for (size\_t i = 0; i < totalStudents; ++i) {

        StudentResult temp;

        temp.ObjectOrientedProgramming = oop[i];

        temp.CalculusAndDifferentialEquations = cald[i];

        temp.LinearAlgebra = la[i];

        temp.CommunicationSkills = comm[i];

        temp.IslamicStudies = isl[i];

        temp.EngineeringDrawing = ed[i];

        mergedResults.push\_back(temp);

    }

    return mergedResults;

}

int main() {

    // Replace this with the actual path to your CSV file

    rapidcsv::Document doc("student\_grades.csv");

    std::vector<int> oopMarks = doc.GetColumn<int>("ObjectOrientedProgramming");

    std::vector<int> caldMarks = doc.GetColumn<int>("CalculusAndDifferentialEquations");

    std::vector<int> edMarks = doc.GetColumn<int>("EngineeringDrawing");

    std::vector<int> laMarks = doc.GetColumn<int>("LinearAlgebra");

    std::vector<int> commSkillsMarks = doc.GetColumn<int>("CommunicationSkills");

    std::vector<int> islamicStudiesMarks = doc.GetColumn<int>("IslamicStudies");

    // Sort each subject's marks

    customSort(oopMarks, 0, oopMarks.size() - 1);

    customSort(caldMarks, 0, caldMarks.size() - 1);

    customSort(edMarks, 0, edMarks.size() - 1);

    customSort(laMarks, 0, laMarks.size() - 1);

    customSort(commSkillsMarks, 0, commSkillsMarks.size() - 1);

    customSort(islamicStudiesMarks, 0, islamicStudiesMarks.size() - 1);

    // Merge and display the sorted subject marks

    std::vector<StudentResult> mergedResults = mergeSubjectMarks(oopMarks, caldMarks, edMarks, laMarks, commSkillsMarks, islamicStudiesMarks);

    for (const auto& student : mergedResults) {

        std::cout << "ObjectOrientedProgramming: " << student.ObjectOrientedProgramming

                  << ", CalculusAndDifferentialEquations: " << student.CalculusAndDifferentialEquations

                  << ", EngineeringDrawing: " << student.EngineeringDrawing

                  << ", LinearAlgebra: " << student.LinearAlgebra

                  << ", CommunicationSkills: " << student.CommunicationSkills

                  << ", IslamicStudies: " << student.IslamicStudies << std::endl;

    }

    return 0;

}

**Output:**

