PROJECT DOCUMENTATION: CGPA Calculator Using C++

Introduction

The CGPA Calculator is a console-based application written in C++. It allows students to calculate their cumulative grade point average (CGPA) and percentage based on their academic performance.

This project aims to improve fundamental programming skills like input handling, loops, functions, and data structures.

Problem Statement

Many students struggle to accurately calculate their CGPA and percentages manually. Manual calculation is time-consuming and error-prone.

There is a need for a simple, quick, and dependable system that can perform these calculations automatically.

This project aims to solve this problem with a C++-based application.

Objectives

The main goals of this project are

To calculate CGPA accurately.

To accept dynamic user input

To validate the entered marks

To display results clearly

To improve programming skills

To apply theoretical knowledge practically

Scope of the Project

The scope of this project includes the following:

A single-user system

console-based interface

Calculation involving multiple subjects

Immediate results display

Tools and Technologies

Programming Language: C++

IDE: Dev-C++

Compiler: GCC

Platform: Windows

System Design

6.1 Architecture

The system follows a simple procedural programming approach.

Main Components:

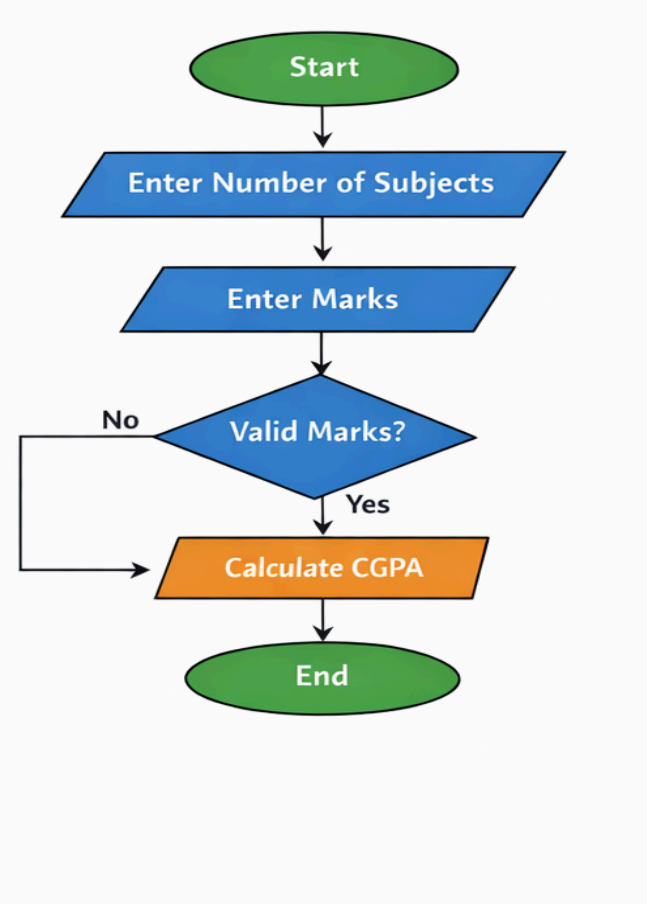
Input Module – Takes marks and subject count

Validation Module – Checks valid range (0–100)

Calculation Module – Computes CGPA

Output Module – Displays results

6.2 Data Flow



Methodology

The development process followed these steps:

1. Requirement analysis
2. Design of algorithm
3. Coding in C++
4. Input validation
5. Testing with sample data
6. Debugging
7. Documentation

Algorithm

1. Start
2. Display welcome message
3. Input number of subjects
4. Validate input
5. Input marks for each subject
6. Validate marks (0–100)
7. Convert marks to grade points
8. Calculate CGPA
9. Calculate percentage
10. Display result
11. End

Implementation Details

The project is implemented using:

- Vectors for storing marks
- Loops for iteration
- Functions for modular coding
- Input validation using conditions
- Output formatting using iomanip

This ensures accuracy and readability.

Project Code

```
#include <iostream>
#include <vector>
#include <iomanip>

using namespace std;

// Calculate CGPA
double calculateCGPA(const vector<double>& marks) {

    double sum = 0;

    for (int i = 0; i < marks.size(); i++) {
        sum += marks[i] / 10.0;
    }

    return sum / marks.size();
}

int main() {

    int n;

    cout << "\n===== CGPA CALCULATOR =====\n";

    // Input number of subjects
    cout << "Enter number of subjects: ";
    cin >> n;

    while (n <= 0) {
        cout << "Invalid number. Try again: ";
        cin >> n;
    }

    vector<double> marks(n);

    // Input marks
    for (int i = 0; i < n; i++) {
```

```
            cout << "Enter marks for subject " << (i + 1) << ": ";
            cin >> marks[i];

            while (marks[i] < 0 || marks[i] > 100) {
                cout << "Invalid marks (0-100). Re-enter: ";
                cin >> marks[i];
            }
        }

        double cgpa = calculateCGPA(marks);
        double percentage = cgpa * 9.5;

        // Output
        cout << fixed << setprecision(2);

        cout << "\n----- RESULT -----\n";
        cout << "CGPA       : " << cgpa << endl;
        cout << "Percentage : " << percentage << "%\n";

        cout << "\nThank you for using CGPA Calculator!\n";

        return 0;
    }
```

How to Run the Project

Step 1: Compile

g++ main.cpp -o cgpa

Step 2: Run

./cgpa

Sample Output

```
===== CGPA CALCULATOR =====
Enter number of subjects: 4
```

```
Enter marks for subject 1: 85
Enter marks for subject 2: 90
Enter marks for subject 3: 78
Enter marks for subject 4: 88

----- RESULT -----
CGPA       : 8.52
Percentage : 80.94%
```

Advantages

- Easy to use
- Fast calculation
- Accurate output
- Beginner-friendly
- No external libraries required

Limitations

- No graphical interface
- No data storage
- Single-user system
- Manual data entry required

Future Enhancements

- Add grade classification
- Semester-wise calculation
- File storage

GUI interface

Web-based version

Learning Outcomes

Through this project, the developer learned:

C++ syntax and structure

Modular programming

Input validation

Problem-solving techniques

Documentation skills

Version control basics

Conclusion

The CGPA Calculator project successfully demonstrates the use of fundamental C++ concepts. It offers an efficient method of calculating academic performance and serves as a solid foundation for advanced projects.

The project met all of the defined objectives and was completed successfully.

References

C++ Official Documentation

GeeksforGeeks

Stack Overflow

Programming Textbooks

Details

Name: **Ponugupati Teja Sri Krishna**
Project Title: CGPA Calculator
Technology: C++
Platform: Windows