

**PROJECT REPORT**

A report submitted in partial fulfillment of the requirements for the



**Project**

School of Computer Science & Engineering By

# 

# Parth Medhekar

PRN No:22SC114281060 Roll No: 47

# Omkar Gurav

PRN No:22SC114281061 Roll No: 48

# Vedant Narote

PRN No:22SC114281063 Roll No: 49

# Vaishnavi Khot

PRN No:22SC114281064 Roll No: 50

Program: BTech Class: SY BTech (Div A)

Under Supervision of

# Mrs.Veena Mali

Academic Year: 2023-2024



School of Computer Science & Engineering

**CERTIFICATE**

This is to certify that the “**Project Report**”

On

# “CGPA CALCULATOR”

submitted by

# Parth Medhekar

PRN No:22SC114281060 Roll No: 47

# Omkar Gurav

PRN No:22SC114281061 Roll No: 48

# Vedant Narote

PRN No:22SC114281063 Roll No: 49

# Vaishnavi Khot

PRN No:22SC114281064 Roll No: 50

Program: BTech CSE Class: SY BTech(Div A)

is work done by him/her and submitted during the 2023 – 2024 academic year, in partial fulfillment of the **Project.**

**Sanjay Ghodawat University, Kolhapur**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mrs. Veena Mali** | **Dr. Deepika Patil** | **Dr. Deepika Patil** |  |
| **Project Guide** | **PBL Co-Ordinator** | **DC CSE &AIML** | **External** |



# DECLARATION

I the undersigned solemnly declare that the report of the project work entitled **“CGPA CALCULATOR”** which is carried out under the supervision of **Mrs. Veena Mali** I assert that the statements made and conclusions drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the project report does not contain any part of any work which has been submitted for the award of any other degree certificate in this University or any other University.

# Student Name:

# Parth Medhekar

PRN No:22SC114281060 Roll No: 47

# Omkar Gurav

PRN No:22SC114281061 Roll No: 48

# Vedant Narote

PRN No:22SC114281063 Roll No: 49

# Vaishnavi Khot

PRN No:22SC114281064 Roll No: 50

# Class: SY BTech (Div A)



# ACKNOWLEDGMENT

First, I would like to thank my Head of the School **Dr. Deepika Patil** for constructive criticism throughout my project. I would like to thank PBL coordinator **Ms. Deepika Patil** and Department Project Guide **Mrs. Veena Mali** for support and advices to get and complete internship in above said organization. It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals. I am extremely grateful to my department staff members and friends who helped me in successful completion of this project.



# ABSTRACT

Cumulative Grade Point Average (CGPA) deals with the maintenance of university, college,faculty, student information within the university. SGS is an automation system, which is used tostore the college, faculty, student, courses and information of a college.Starting from registration of a new student in the college, it maintains all the details regarding theattendance and marks of the students. The project deals with retrieval of information through an INTRANET based campus wide portal. It collects related information from all the departmentsof an organization and maintains files, which are used to generate reports in various forms tomeasure individual and overall performance of the students.Development process of the system starts with System analysis. System analysis involvescreating a formal model of the problem to be solved by understanding requirements

# TABLE OF CONTENT’S

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **Title** | **Page No.** |
| 1 | Introduction | 1 |
| 2. | Objective | 5 |
| 3 | System Requirements Specification(SRS) | 6 |
| 4 | Methodology | 7 |
| 5 | Implementation | 9 |
| 6 | Result | 20 |
| 7 | Conclusion And Future Scope | 21 |
| 8 | References | 23 |

# Introduction

## Introduction

The "Cumulative Grade Point (CGP) Calculator" is a C++ program designed to assist students in calculating their cumulative grade point average the program prompts the user to input the number of subjects they are enrolled in For each subject, the user is asked to enter the credit hours associated with the subject and the corresponding grade obtained on a scale of 0 to 4 The program.

Users can use our CGP calculator to tackle various problem types, including symbolic regression, logical circuit design, and more. By providing an appropriate target function and configuration, the calculator adapts to the specific problem at hand.Our C++ implementation is designed with modularitly in mind, making it easy for users to extend and customize the calculator for their specific needs.

In conclusion, our CGP calculator in C++ is a versatile tool for solving a wide range of optimization and design problems. It leverages the power of evolutionary algorithms and CGP to explore solution spaces efficiently. Whether you're a researcher, engineer, or enthusiast, our calculator can help you tackle complex problems with ease.

## Problem Definition:

Your code should be designed to calculate the cumulative GPA (CGPA) for any students. The

code should work for any number of semesters selected by the user, and for different number of

courses taken in each semester.

The code should begin by asking the user for the number of semesters that he has completed,

then number of courses taken in each semester, and based on that, the user will start to enter his

grade in cach course along with the number of credits for that course.

The CGPA calculator should have an accurate mapping of grades to grade points. If this mapping is incorrect, it can lead to inaccurate CGPA calculations. Ensure that the grading system is properly defined and updated.

## Scope:

The scope of the project is to calculate CGPA easily and it is efficient for student. Whom want to calculate his CGPA just input his number then this program automatically calculates his CGPA oe GPA and show him.

1. Educational Level: Determine whether the CGPA calculator is designed for high school, undergraduate, graduate, or postgraduate students. Different educational levels may have varying grading systems and policies that need to be accommodated.
2. Grading System: Decide which grading systems the CGPA calculator will support, such as the 4.0 scale, 5.0 scale, or any other institution-specific grading system. Ensure that it can handle letter grades, numerical grades, or other grading formats.
3. User Interface: Define the user interface and user experience to make the calculator user-friendly, including clear instructions, easy data input, and the ability to add multiple courses and semesters.
4. User Feedback: Establish mechanisms for collecting user feedback to continuously improve the calculator and address any issues or suggestions from users.

## Problem Identification:

Cumulative Grade Point Average System using C++ could address includes:

1. Missing or Incorrect Course Data: Users may input incorrect course data, such as incorrect grades or credit hours, leading to inaccurate CGPA calculations. Ensure data validation to prevent such issues.

2. Inconsistent Grading Systems: Different educational institutions may use different grading systems. A CGPA calculator should be flexible enough to handle various grading systems and scales.

3. Incorrect Formula: The CGPA calculator should use the correct formula to calculate the CGPA. Ensure that the formula used for the calculation aligns with your educational institution's policies.

4. Lack of User-Friendly Interface: A complex or confusing user interface can lead to input errors. The calculator should be easy to use and provide clear instructions.

5. Compatibility Issues: Ensure that the CGPA calculator is compatible with various devices and web browsers to prevent technical issues.

# Objectives

1. The CGPA Calculator in C++ is a console-based application designed to help students calculate and manage their Cumulative Grade Point Average (CGPA). This program allows users to input their course grades and credits and then computes the CGPA based on inputs.
2. The program provides a user-friendly interface where students can input their course grades and credits.
3. The application includes input validation to ensure that users enter valid grades and credits. It prompts the user to re-enter data if invalid inputs are provided.
4. The application includes input validation to ensure that users enter valid grades and credits. It prompts the user to re-enter data if invalid inputs are provided.
5. Once the user has inputted all their course grades and credits, the program calculates the CGPA by applying the standard CGPA formula.
6. After successfully entering the course details and computing the CGPA, the program displays the calculated CGPA to the user.
7. The program allows users to exit and restart the application, maintaining their data until they decide to reset it.

# System Requirements Specification

## Software Requirement:

* Microsoft Visual Studio Code
* Dev C++

## Hardware Requirement

* 4GB RAM
* 500 GB HDD
* o i3 processor

# Methodology

## Algorithm:

## Step 1 : Start

## Step 2 : Choose the options for calculating CGPA

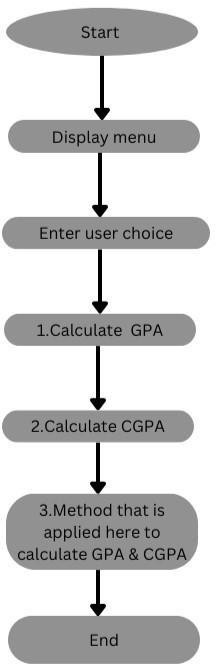
## Step 3: Enter number of subjects

## Step 4 : Enter credits for subjects

## Step 5 : Displays CGPA

## Step 6 : End

* **Flow Diagram(Flow Chart):**



# Implementation

1. A CGPA calculator using C++ would include the ability to input course grades and credit hours, calculate the weighted grade points for each course,compute the Cumulative Grade Point Average (CGPA) based on the accumulated creditsand grade points, and display the final CGPA to the user.
2. Additionally, the calculator should handle different grade scales and provide errorhandling for invalid inputs.
3. CGPA calculator using C++ should allow users to input their course grades and corresponding credit hours.
4. he program should then calculate the grade points for each course based on the grade scale (e.g., A, B, C, etc.), and then compute the CGPA by taking the sum of (grade points \*credit hours) for all courses and dividing it by the total credit hours.
5. Error handling should be in place to handle invalid inputs and ensure accuratecalculations. Finally, the calculated CGPA should be displayed to the user.

****

**“Introduction of C++Programming”**

**C**++ is a powerful and versatile programming language that originated as an extension of the C programming language. Developed by Bjarne Stroustrup in the early 1980s, C++ is known for its efficiency, flexibility, and high performance. It's widely used in various domains such as system software, application software, game development, embedded systems, and more.

### Key Aspects of C++:

1. \*Object-Oriented Programming (OOP):\* C++ supports the key principles of OOP, including classes, objects, inheritance, polymorphism, and encapsulation. This paradigm enables developers to organize and structure their code more efficiently.

2. \*Efficiency and Performance:\* C++ allows low-level memory manipulation, providing greater control over system resources. This makes it well-suited for performance-critical applications like games or operating systems.

3. \*Standard Template Library (STL):\* The STL offers a collection of classes and functions for common programming tasks, like data structures, algorithms, and I/O operations. It includes containers (vectors, lists, maps), algorithms, and iterators, enhancing code reusability and efficiency.

4. \*Portability:\* C++ code can be compiled and executed on various platforms, offering portability across different operating systems and hardware.

5. \*Wide Applicability:\* It's used in a variety of fields, from developing system software (like operating systems) to creating high-performance applications, games, and even in embedded systems.

6. \*Support for Modern Features:\* With each new standard (C++11, C++14, C++17, C++20), C++ continues to evolve, introducing modern features like lambda expressions, smart pointers, multithreading support, and more.

**Advantages of learning C++**:

Learning C++ offers numerous advantages due to its versatility, performance, and wide range of applications:

1. Performance and Efficiency:

\*Speed:\* C++ allows direct manipulation of hardware resources, enabling efficient code execution.

\*Low-level Manipulation:\* Provides control over memory allocation and management, making it highly efficient.

2. Versatility:

\*Multi-paradigm Language:\* Supports procedural, object-oriented, and generic programming, offering a flexible approach to problem-solving.

\*Large Standard Library:\* The Standard Template Library (STL) includes containers, algorithms, and utilities for various programming tasks.

3. Wide Applicability:

\*System Software:\* Used in developing operating systems, device drivers, and other system software.

\*Application Software:\* Employed in building high-performance applications, games, and complex software systems.

\*Embedded Systems:\* Commonly used in embedded systems and IoT devices where resource efficiency is crucial.

4. Career Opportunities:

\*In-Demand Skill:\* C++ remains a sought-after language in industries like game development, finance, high-performance computing, and more.

\*Foundation for Learning Other Languages:\* Understanding C++ provides a strong foundation for learning other languages due to its comprehensive nature.

5. Strong Community and Resources:

\*Vast Community Support:\* A large community of C++ developers provides access to resources, forums, and open-source projects.

\*Abundance of Tools and Libraries:\* Numerous tools and libraries are available to facilitate development in C++.

6. Critical Thinking and Problem-Solving:

\*Memory Management:\* Learning C++ sharpens understanding of memory allocation, pointers, and resource management, promoting a deeper understanding of computer architecture.

\*Optimized Solutions:\* Challenges in C++ often require critical thinking and efficient problem-solving, fostering valuable skills applicable across different domains.

## Source Code:

## #include <iostream>

## #include <stdlib.h>

## using namespace std;

## void calculateGPA();

## void calculateCGPA();

## void method();

## int main()

## {

## system("cls");

## int input;

## cout<<"--------------------------------------------------------------------------"<<endl;

## cout<<" GPA & CGPA Calculator "<<endl;

## cout<<"--------------------------------------------------------------------------\n"<<endl;

## cout<<" MENU:"<<endl;

## cout<<" 1. Calculate GPA (Grade Point Average)"<<endl;

## cout<<" 2. Calculate CGPA (Cummulative Grade Point Average)"<<endl;

## cout<<" 3. Method that is applied here for calclating GPA & CGPA"<<endl;

## cout<<" 4. Exit Application"<<endl;

## cout<<"--------------------------------------------------------------------------"<<endl;

## sub:

## cout<<"Enter your choice: ";

## cin>>input;

## switch(input)

## {

## case 1:

## calculateGPA();

## break;

## case 2:

## calculateCGPA();

## break;

## case 3:

## method();

## break;

## case 4:

## exit(EXIT\_SUCCESS);

## break;

## default:

## cout<<"You have entered wrong input.Try again!\n"<<endl;

## goto sub;

## break;

## }

## }

## void calculateGPA()

## {

## int q;

## system("cls");

## cout<<"-------------- GPA Calculating -----------------"<<endl;

## cout<<" How many subject's points do you want to calculate? : ";

## cin>>q;

## float credit [q];

## float point [q];

## cout<<endl;

## for(int i=0;i<q;i++)

## {

## cout<<"Enter the credit for the subject "<<i+1<<": ";

## cin>>credit[i];

## cout<<endl;

## cout<<"Enter the point of the subject "<<i+1<<": ";

## cin>>point[i];

## cout<<"-----------------------------------\n\n"<<endl;

## }

## float sum=0;

## float tot;

## for(int j=0;j<q;j++)

## {

## tot=credit[j]\*point[j];

## sum=sum+tot;

## }

## float totCr=0;

## for(int k=0;k<q;k++)

## {

## totCr=totCr+credit[k];

## }

## cout<<"\n\n\nTotal Points: "<<sum<<" . Total Credits: "<<totCr<<" .Total GPA: "<<sum/totCr<<" ."<<endl;

## sub:

## int inmenu;

## cout<<"\n\n\n1. Calculate Again"<<endl;

## cout<<"2. Go Back to Main Menu"<<endl;

## cout<<"3. Exit This App \n\n"<<endl;

## cout<<"Your Input: "<<endl;

## cin>>inmenu;

## switch(inmenu)

## {

## case 1:

## calculateGPA();

## break;

## case 2:

## main();

## break;

## case 3:

## exit(EXIT\_SUCCESS);

## default:

## cout<<"\n\nYou have Entered Wrong Input!Please Choose Again!"<<endl;

## goto sub;

## }

## }

## void calculateCGPA()

## {

## system("cls");

## int l;

## cout<<"-------------- CGPA Calculating -----------------\n\n"<<endl;

## cout<<"How many semester results do you want input? :";

## cin>>l;

## cout<<"\n\n"<<endl;

## float semrs[l];

## int i;

## for(i=0;i<l;i++)

## {

## cout<<" Enter Semester "<<i+1<<" Result(GPA): ";

## cin>>semrs[i];

## cout<<"\n"<<endl;

## }

## float semtot=0;

## for(int j=0;j<l;j++)

## {

## semtot=semtot+semrs[j];

## }

## cout<<"\*\*\* Your CGPA is "<<semtot/l<<" \*\*\*\*\*"<<endl;

## sub:

## int inmenu;

## cout<<"\n\n\n1. Calculate Again"<<endl;

## cout<<"2. Go Back to Main Menu"<<endl;

## cout<<"3. Exit This App \n\n"<<endl;

## cout<<"Your Input: "<<endl;

## cin>>inmenu;

## switch(inmenu)

## {

## case 1:

## calculateCGPA();

## break;

## case 2:

## main();

## break;

## case 3:

## exit(EXIT\_SUCCESS);

## default:

## cout<<"\n\nYou have Entered Wrong Input!Please Choose Again!"<<endl;

## goto sub;

## }

## }

## void method()

## {

## system("cls");

## cout<<"--------------- Method of Calculating GPA & CGPA ---------------\n\n"<<endl;

## cout<<" GPA= Sum of (Credit\*Point) / total of credits \n"<<endl;

## cout<<" CGPA= Sum of GPA / number of semesters "<<endl;

## cout<<"-----------------------------------------------------------------\n\n"<<endl;

## sub:

## int inmenu;

## cout<<"1. Go Back to Main Menu"<<endl;

## cout<<"2. Exit This App \n\n"<<endl;

## cout<<"Your Input: "<<endl;

## cin>>inmenu;

## switch(inmenu)

## {

## case 1:

## main();

## break;

## case 2:

## exit(EXIT\_SUCCESS);

## default:

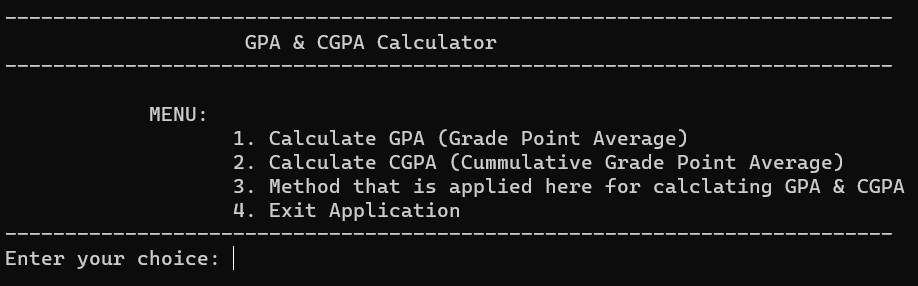
## cout<<"\n\nYou have Entered Wrong Input!Please Choose Again!"<<endl;

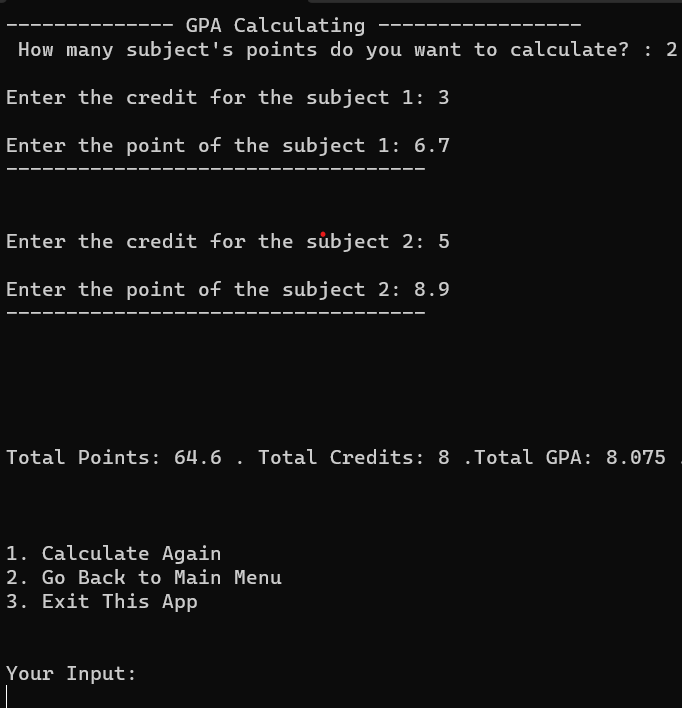
## goto sub;

## }

## };

# Result





# Conclusion & Future Scope

## Future Scope:

The scope of this research is to explore those ways through which GPA can be calculated easily.

The knowledge will be gatheredthrough the help of several secondary andprimary

sources. The project will provide a proper framework of GPA calculator. The final result will be

extracted from the master recordof student data. It will also show the student attendance status.

Moreover, the graph will also be presented in order to show student‟s performance

.

## 1. \*Industry Relevance:\* Cumulative Grade Point Average systems are widely used in educational institutions. Developing this system in C++ can provide a solid understanding of system development that's still relevant in the industry.

## 2. \*Software Development Skills:\* Building a CGPA system allows you to enhance your skills in C++ programming, data structures, algorithms, and system design, which are valuable skills in software development.

## 3. \*Career Prospects:\* Knowledge in C++ and system development can open doors for roles in software development, system architecture, database management, and more.

## 4. \*Entrepreneurial Opportunities:\* You could potentially market and sell your CGPA system or use it as a foundation for other software ventures.

## 5. \*Enhanced Problem-solving Abilities:\* Developing such a system can improve your problem-solving skills and logical thinking, valuable for tackling complex challenges in various domains.

## Conclusion:

In conclusion, a CGPA (Cumulative Grade Point Average) calculator is an indispensable tool in the academic world, providing students, educators, and institutions with a reliable means of assessing and understanding academic performance. The project titled as CGPA was deeply studied and analyzed to design the code and implement. A user-friendly interface, effective error handling, and a commitment to privacy and security are essential features that enhance the calculator's utility. Overall, the CGPA calculator simplifies academic record-keeping and decision-making, contributing significantly to the academic success of students and the efficiency of educational institutions.

# References

## BOOKS

1. C++ Primer.
2. Effective Modern C++.
3. Professional C++.
4. A Tour of C++.

## Links

## https://gist.github.com

* 1. [https://www.studocu.com](https://www.studocu.com/)
  2. https://[www.quora.com](http://www.quora.com/)
  3. <https://www.geeksforgeeks.org>