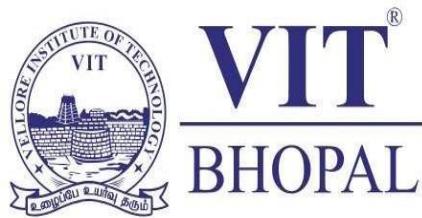


VITYARTHI – PROJECT



Project title : VIT'ian GPA and CGPA calculator

Course title : Introduction to problem solving and programming

Course code : CSE 1021

Course type : Flipped course

Course credits : 4

Professor : Dr. Sasmita Padhy

Slot : C14 + E11 + E12

Submitted by : Raghul.S.K

Registration number : 25BAI10288

Submission date : 21/11/2025

VITYARTHI – PROJECT

INTRODUCTION :

The “VIT’ian GPA and CGPA Calculator” is simple but an effective tool designed to help students like us to calculate our academic performance. The Cumulative Grade Point Average (CGPA) is a widely used benchmark to evaluate a student’s overall academic performance and it is essential for students to track their progress throughout their academic journey. The “VIT’ian GPA and CGPA Calculator” is designed to provide a platform for the students where they enter their marks for each semester to calculate their GPA and CGPA.

The calculator will take all the required inputs into account and then it will provide an accurate calculation of the GPA and CGPA. The “VIT’ian GPA and CGPA Calculator” is developed using Python, a programming language known for it’s easy to learn, versatility and extensive libraries. The calculator is designed to be easy to use for the students, with simple and clear instructions to guide the students of all levels in calculating GPA and CGPA with their own marks.

The “VIT’ian GPA and CGPA calculator” is a project that demonstrates the application of programming knowledge in python programming language I have learnt throughout this semester in solving a real-world problem. I hope this calculator will be helpful for all the students who want to keep track of their progress throughout their academic journey.

VITYARTHI – PROJECT

PROBLEM STATEMENT :

We, the students of VIT often face difficulties in understanding and manually calculating our semester GPA and CGPA due to diverse assessments and numerous subjects in a particular semester. Without VTOP calculating our GPA and CGPA would be a hectic job, so I have created this “VIT’ian GPA and CGPA Calculator” as the solution for this trouble. The aim of this project is to create a calculator that takes the relative grading into account and delivers instant and reliable results.

PROJECT OBJECTIVES :

- Creating a GPA and CGPA calculator for the students to keep track of their academic progress.
- Applying the programming concepts such as functions, loops, Data types, etc learnt from this course in creation of the calculator.
- The calculator must be easy and simple for the students to use and calculate their GPA and CGPA.
- The calculator must ensure accuracy and reliability in the calculated GPA and CGPA.
- It must provide a user-friendly output that displays the GPA and CGPA of the student.
- Testing and validating the calculator with sample data to ensure its working ability, accuracy and reliability.

VITYARTHI – PROJECT

FUNCTIONAL REQUIREMENTS:

1. User-System Interaction :

- The calculator provides a user-friendly environment for the user to calculate their grades.
- The instructions are simplified and made that it will guide the student throughout the process.
- The system asks the user after each and every semester's GPA calculation whether to continue the process or exit with the current CGPA.

2. Data Input :

- The system allows the user to enter their marks for each semester separately
- The system allows the user to choose the type of the course (LT/LTP/PJ) and enter their credits respectively.
- The program is built only using a single function.

3. Data processing :

• Grade mapping :

With the help of the entered marks and attendance percentage the system maps the grade with respect to the class average and standard deviation of the class (Relative Grading).

• GPA calculation :

After assigning each grade to the courses according to the inputs the system calculates the Grade Point Average (GPA) using the formula,

$$\text{GPA} = \frac{\sum(\text{Course credits} \times \text{Grade points})}{\text{Total credits for current semester}}$$

VITYARTHI – PROJECT

- CGPA calculation :

Once after the GPA is calculated for the particular semester the system calculates the CGPA is calculated using the formula,

$$\text{CGPA} = \frac{\sum(\text{GPA} \times \text{Total credits of particular semester})}{\text{Total credits completed till date}}$$

- Multi-Semester support :

The system shall allow the user to input marks and calculate Grades for multiple semesters or exit after a particular semester.

- CGPA rounding :

The system rounds off the calculated CGPA to 2 decimal places to simplify it and make it easier to work with.

4. Output :

- The System displays the Course credits, Grade acquired, its respective grade points along with the semester wise GPA and CGPA.
- The system checks for the attendance percentage and marks if it is below 75% and 40% respectively it automatically sets the grade to F with 0 grade points.

5. Data storage :

- The input data entered by user is stored by the system for the future reference and future calculation of the Grades and Cumulative Grade Point Average. And the CGPA is manipulated after calculation of each semester's GPA.

VITYARTHI – PROJECT

NON-FUNCTIONAL REQUIREMENTS :

1. Performance :

- The system shall respond with the GPA and CGPA to the user's inputs within a couple seconds.
- The system shall handle a large number of the inputs entered by the user.

2. Security :

- The system will not hold / store any sensitive data of the user.
- The system shall be secure against all the common web vulnerabilities.

3. Usability :

- The system is made to be simple and easy for the user to use and the set of instruction will guide the user throughout their calculation process.

4. Reliability :

- The system shall be available anytime the user wants to calculate their GPA and CGPA.
- And the system can be accessed anywhere around the world with a python code executer.

5. Maintainability :

- The system is written with a maintainable and understandable code with comment lines.
- The system has proper documentation so that it can be updated in the coming future.

VITYARTHI – PROJECT

6. Error handling :

- If the system detects any error in the inputs entered by the user it displays an error message and requests' the user to try entering the inputs again.

7. Resource efficiency :

- The system is designed to minimise power consumption and optimize battery life.
- The system shall require less than 10MB of storage space for optimised running.
- The system doesn't require any network bandwidth so it can be accessed even in places without proper internet connection.

SYSTEM ARCHITECTURE :

- Presentation layer is built using Jupyter Notebook, where the inputs are accepted from the user.
- Application layer is where the input data is processed and the calculations are performed then the outputs are stored and displayed later.
- Data layer is a storage space where the user inputs and the output values are stored.
- Output renderer is a dynamic area that shows all results including detailed mark to grade mapping, GPA/CGPA and alerts on errors.

VITYARTHI – PROJECT

DESIGN DIAGRAM :

1. Workflow :

User input → Data validation → Mark calculation(MTE, TEE, Internals, Attendance) → Grade mapping → GPA Calculation → CGPA aggregation → Result display.

2. Data flow Diagram :

- Course details :

The system allows the user to enter type, credits and marks (MTE, TEE, Internals, Attendance) for each course.

- Validation :

The system after entering the inputs checks for the correctness of inputs such as : “marks within possible ranges”, “Attendance below 100%”.

- Computation:

The system calculates total marks, then marks to the respective grade and grade points for each course separately.

- Result aggregation :

For semester's GPA = $\frac{\sum(\text{Grade points} \times \text{Credits})}{\text{Total credits for current semester}}$

And CGPA is calculated using,

CGPA = $\frac{\sum(\text{GPA} \times \text{Total credits of that semester})}{\text{Total semester completed till date}}$

- Output :

Displays the processed data, the results and then allows the user to calculate GPA and CGPA for the consecutive semesters.

VITYARTHI – PROJECT

3. Component diagram :

- Input form : Collects course details from the user
- Validation module : Checks all input ranges and types
- Calculation core : Computes grade, GPA, CGPA
- Output display : Presents the results

DESIGN DECISIONS & RATIONALE :

1. Programming language : Python

- Rationale : Python is a simple, easy to learn language with large number of libraries and frameworks, making it idle for small projects like this.

2. User interface ; Jupyter Notebook

- Rationale : A simple UI is sufficient for this program so Jupyter is chosen which provides an interactive environment for user to input data and view results.

3. Data storage : In-memory data storage

- Rationale : Since the project doesn't require any explicit data storage, in-memory storage is sufficient and efficient.

4. Calculation Algorithm : Simple formula-based calculation

- Rationale :The GPA and CGPA calculation formula is straight forward and is simple to use.

5. Error handling :

- Rationale : Basic error handling is sufficient for the small project like this and the user is expected to enter valid data.

VITYARTHI – PROJECT

IMPLEMENTATION DETAILS :

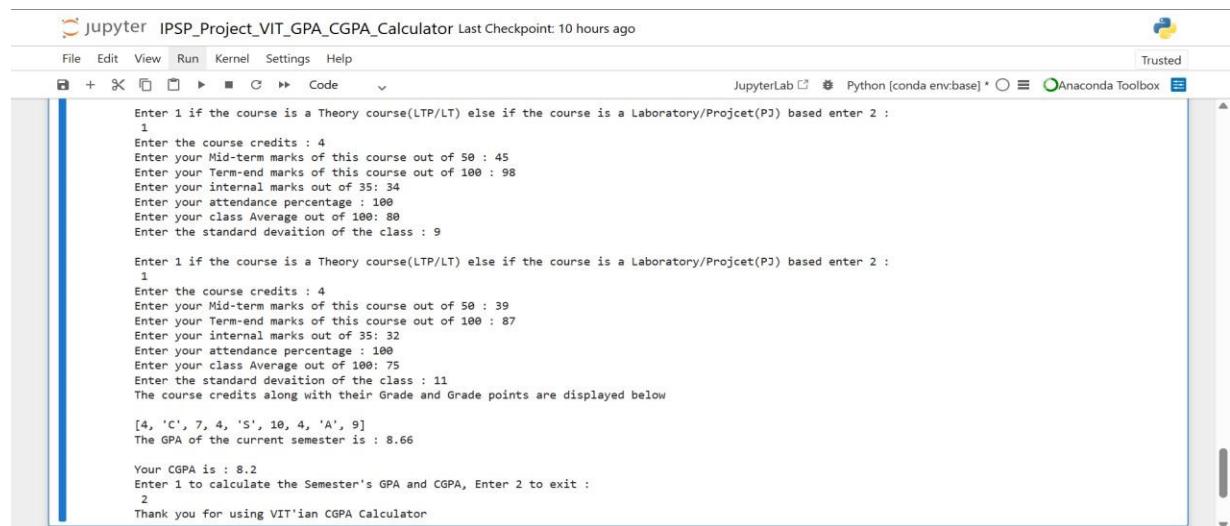
1. Language : Python
2. Development environment : Jupyter Notebook
3. Module : IPSP_Project_VIT_GPA_CGPA_Calculator
4. Function : GPA_calculator
5. Implementation steps :
 - Implement the GPA_calculator function to calculate GPA and CGPA based on input marks and attendance.
 - Once after it is implemented the function itself prints the GPA and CGPA of the user without being called again.
 - And it allows user to calculate for multiple semesters.
 - Test the program with sample inputs and verify the results.

TESTING APPROACH :

1. Unit tests :
Manually validated the system's logic using known correct GPAs from academic records and mock data.
2. System Integration :
Tested with full semester data for all possible scenarios such as Highest(S), lowest(F) grade performance and 0 attendance.
3. Validation :
Entered the values at the boundary, the values such as exactly at grade breakpoints.
4. User flow testing :
Ensure that an error message display if any of the input is wrong.
5. Peer view :
Verified among my peer group on calculations and reliability.

VITYARTHI – PROJECT

SCREENSHOTS / RESULTS :

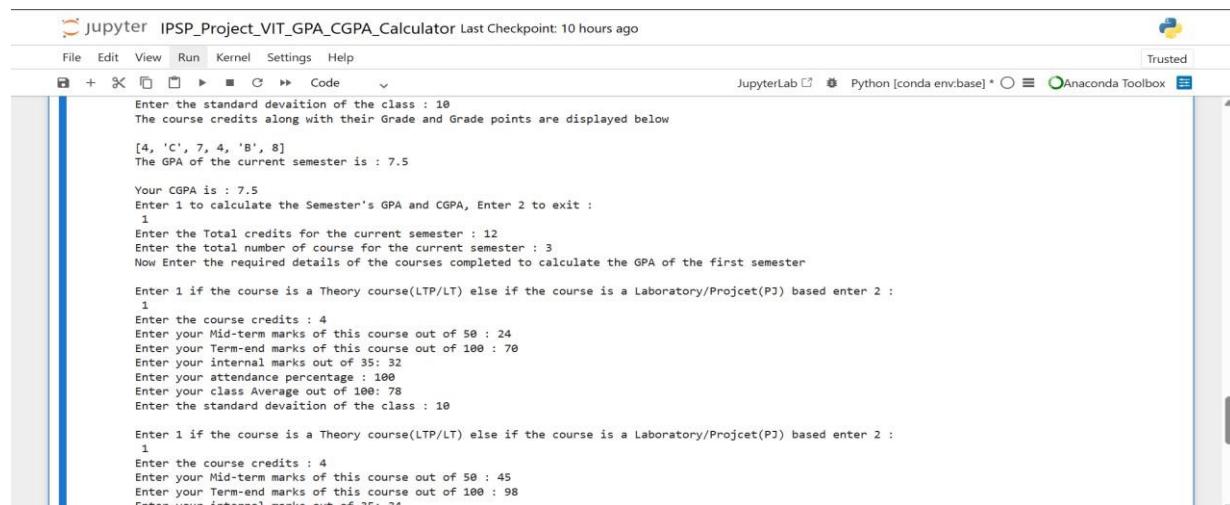


```
jupyter IPSP_Project_VIT_GPA_CGPA_Calculator Last Checkpoint: 10 hours ago
File Edit View Run Kernel Settings Help
JupyterLab Python [conda env:base] * Anaconda Toolbox Trusted
+ ☒ □ ▶ ■ C ▶ Code v
Enter 1 if the course is a Theory course(LTP/LT) else if the course is a Laboratory/Project(PJ) based enter 2 :
1
Enter the course credits : 4
Enter your Mid-term marks of this course out of 50 : 45
Enter your Term-end marks of this course out of 100 : 98
Enter your internal marks out of 35: 34
Enter your attendance percentage : 100
Enter your class Average out of 100: 80
Enter the standard deviation of the class : 9

Enter 1 if the course is a Theory course(LTP/LT) else if the course is a Laboratory/Project(PJ) based enter 2 :
1
Enter the course credits : 4
Enter your Mid-term marks of this course out of 50 : 39
Enter your Term-end marks of this course out of 100 : 87
Enter your internal marks out of 35: 32
Enter your attendance percentage : 100
Enter your class Average out of 100: 75
Enter the standard deviation of the class : 11
The course credits along with their Grade and Grade points are displayed below

[4, 'C', 7, 4, 'S', 10, 4, 'A', 9]
The GPA of the current semester is : 8.66

Your CGPA is : 8.2
Enter 1 to calculate the Semester's GPA and CGPA, Enter 2 to exit :
2
Thank you for using VIT'ian CGPA Calculator
```



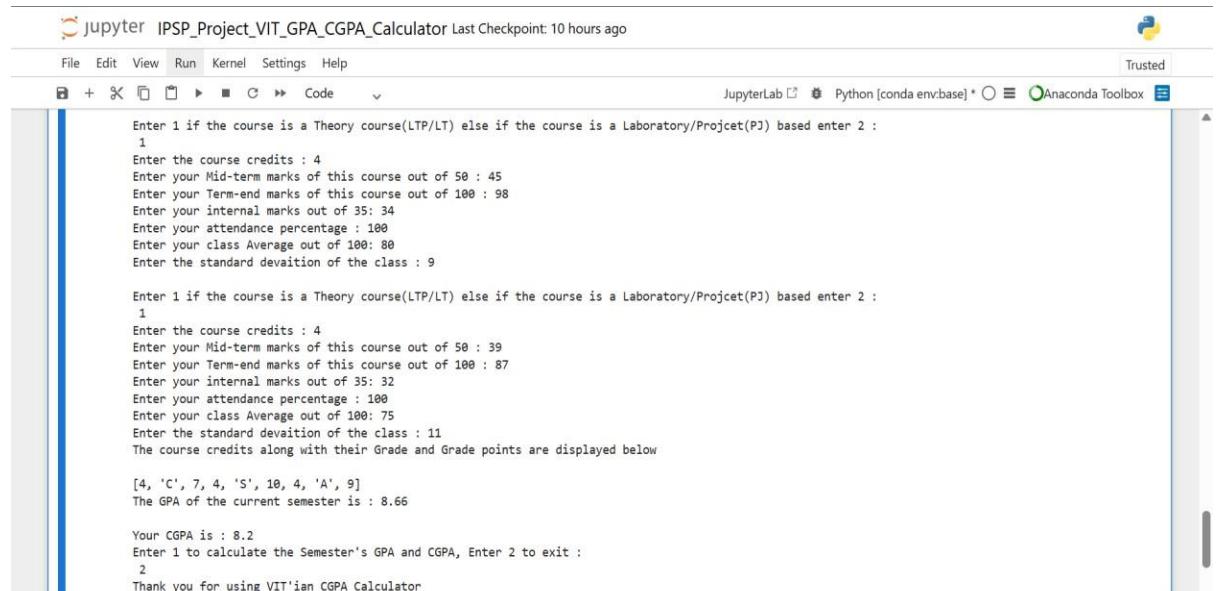
```
jupyter IPSP_Project_VIT_GPA_CGPA_Calculator Last Checkpoint: 10 hours ago
File Edit View Run Kernel Settings Help
JupyterLab Python [conda env:base] * Anaconda Toolbox Trusted
+ ☒ □ ▶ ■ C ▶ Code v
Enter the standard deviation of the class : 10
The course credits along with their Grade and Grade points are displayed below

[4, 'C', 7, 4, 'B', 8]
The GPA of the current semester is : 7.5

Your CGPA is : 7.5
Enter 1 to calculate the Semester's GPA and CGPA, Enter 2 to exit :
1
Enter the Total credits for the current semester : 12
Enter the total number of course for the current semester : 3
Now Enter the required details of the courses completed to calculate the GPA of the first semester

Enter 1 if the course is a Theory course(LTP/LT) else if the course is a Laboratory/Project(PJ) based enter 2 :
1
Enter the course credits : 4
Enter your Mid-term marks of this course out of 50 : 24
Enter your Term-end marks of this course out of 100 : 70
Enter your internal marks out of 35: 32
Enter your attendance percentage : 100
Enter your class Average out of 100: 78
Enter the standard deviation of the class : 10

Enter 1 if the course is a Theory course(LTP/LT) else if the course is a Laboratory/Project(PJ) based enter 2 :
1
Enter the course credits : 4
Enter your Mid-term marks of this course out of 50 : 45
Enter your Term-end marks of this course out of 100 : 98
Enter your internal marks out of 35: 34
```



```
jupyter IPSP_Project_VIT_GPA_CGPA_Calculator Last Checkpoint: 10 hours ago
File Edit View Run Kernel Settings Help
JupyterLab Python [conda env:base] * Anaconda Toolbox Trusted
+ ☒ □ ▶ ■ C ▶ Code v
Enter 1 if the course is a Theory course(LTP/LT) else if the course is a Laboratory/Project(PJ) based enter 2 :
1
Enter the course credits : 4
Enter your Mid-term marks of this course out of 50 : 45
Enter your Term-end marks of this course out of 100 : 98
Enter your internal marks out of 35: 34
Enter your attendance percentage : 100
Enter your class Average out of 100: 80
Enter the standard deviation of the class : 9

Enter 1 if the course is a Theory course(LTP/LT) else if the course is a Laboratory/Project(PJ) based enter 2 :
1
Enter the course credits : 4
Enter your Mid-term marks of this course out of 50 : 39
Enter your Term-end marks of this course out of 100 : 87
Enter your internal marks out of 35: 32
Enter your attendance percentage : 100
Enter your class Average out of 100: 75
Enter the standard deviation of the class : 11
The course credits along with their Grade and Grade points are displayed below

[4, 'C', 7, 4, 'S', 10, 4, 'A', 9]
The GPA of the current semester is : 8.66

Your CGPA is : 8.2
Enter 1 to calculate the Semester's GPA and CGPA, Enter 2 to exit :
2
Thank you for using VIT'ian CGPA Calculator
```

VITYARTHI – PROJECT

CHALLENGES FACED :

1. Relative grading :

VIT's relative grading for theory courses took me a long time to understand but I researched a lot on it then found the logic behind it and applied it in my program.

2. User error handling :

Invalid inputs such as marks > 100, attendance percentage above 100 always makes the calculation so wrong so checking on them was quite a hard task.

3. Dynamic update :

Managing the current session CGPA calculation as the semester are added one by one required great care and attention for proper management

4. Transparency :

Providing the user with the clear step by step Calculations and not just their final results.

LEARNING AND KEY TAKEAWAYS :

1. Implementation :

Learnt how to implement the acquired knowledge to solve the real-world problems, acquired a lot of practical skills in shape-shifting the academics into a key for locks of problem in today's world.

2. Automation value :

This would have took a lot of time and energy if it was done manually but now within seconds with the help of this calculator we can get the results automatically in a few seconds.

VITYARTHI – PROJECT

3. Testing :

Coding part is easy if we know the logic behind it but we can only know our code is correct when it test run it with multiple values, so I understood that testing a system is as important as designing a system.

4. Communication :

The system's ability of guidance has been improved and error messages are directly displayed in the output UI for optimal user support.

FUTURE ENHANCEMENTS :

1. Data persistence :

Add a support for local storage, so that the user can store his previous session data and prevent it from losing in the next session.

2. User Interface :

Separately designing a new User Interface (App / Webpage) for better conditions and presentation and instant access of the system rather than running it in Jupyter Notebook.

3. Analytics :

With the help of the previous data entered by the user predicting their next most probable GPA and CGPA and creating graphical dashboards for easier access.

4. Export :

Enabling exporting reports as in the form of PDF or EXCEL or Direct email share formats for the student's use.

VITYARTHI – PROJECT

REFERENCE :

1. VIT BHOPAL Academic guidelines for Grading and Credit calculation.
2. VITYARTHI and CSE1021 course for Python coding skills.
3. Google for learning about Documentation.
4. Peer and mentor feedbacks from,
 - Dr. Sasmita Padhy,
 - Gokul – 22BCG10045,
 - Gopi – 23BSA10162,
 - Gokul Prasad. K - 25BAI10148,
 - Bavesh. P – 25BAI10093,
 - Sudharshan. SG – 25BAI10706,
 - Dilip. S – 25BAI10590

PROGRAM CODE :

```
## Creating a CGPA calculator for VIT'ans print("Welcome  
to VIT'ian CGPA CALCULATOR\n") print("Let's now calculate  
the first semester's GPA\n") def GPA_Calculator():  
    Total_credits=int(input("Enter the Total credits for the current  
semester :"))  
    Number_of_courses=int(input("Enter the total number of course for the  
current semester :"))  
    print("Now Enter the required details of the courses completed to  
calculate the GPA of the first semester")  
    list_courses=[]  
    list_grades=[]  
    GPA=0  
    for i in range(0,Number_of_courses):  
        print()  
        Type_of_course=int(input("Enter 1 if the course is a Theory  
course(LTP/LT) else if the course is a Laboratory/Project(PJ) based  
enter 2 :\n"))  
        Course_credit=int(input("Enter the course credits :"))  
        if(Type_of_course==1):  
            MTE=float(input("Enter your Mid-term marks of this course out  
of 50 :"))  
            TEE=float(input("Enter your Term-end marks of this course out  
of 100 :"))  
            Internals=float(input("Enter your internal marks out of  
35:"))  
            Attendance_percentage=float(input("Enter your attendance  
percentage :"))  
            Class_avg=float(input("Enter your class Average out of  
100:"))  
            SD=float(input("Enter the standard deviation of the  
class :"))  
            if(MTE <= 50 and TEE <= 100 and Internals <= 35 and  
Attendance_percentage <= 100 and Class_avg <= 100):  
                #Calculation TEE and MTE marks according to their  
                weightage  
                MTE=(MTE/50)*30  
                TEE=(TEE/100)*30  
                #Calculation of Attendance marks  
                if(Attendance_percentage>=96):  
                    Attendance_marks=5  
                elif(Attendance_percentage>=91 and  
Attendance_percentage<=95):  
                    Attendance_marks=4  
                elif(Attendance_percentage>=86 and  
Attendance_percentage<=90):  
                    Attendance_marks=3  
                elif(Attendance_percentage>=81 and  
Attendance_percentage<=85):
```

```

        Attendance_marks=2
    elif(Attendance_percentage>=75 and
Attendance_percentage<=80):
        Attendance_marks=1
    else:
        Attendance_marks=0
    Total_marks=MTE+TEE+Internals+Attendance_marks
    #Calculation of grade
    if(Attendance_marks==0):
        Grade="F"
        Grade_points=0
    else:
        if(Total_marks>(Class_avg+(1.50*SD))):
            Grade="S"
            Grade_points=10
        elif(Total_marks>(Class_avg+(0.50*SD)) and
(Total_marks<=(Class_avg+(1.50*SD))) :
            Grade="A"
            Grade_points=9
        elif(Total_marks>(Class_avg-(0.50*SD)) and
(Total_marks<=(Class_avg+(0.50*SD))) :
            Grade="B"
            Grade_points=8
        elif(Total_marks>(Class_avg-(1.0*SD)) and
(Total_marks<=(Class_avg-(0.50*SD))) :
            Grade="C"
            Grade_points=7
        elif(Total_marks>(Class_avg-(1.50*SD)) and
(Total_marks<=(Class_avg-(1.0*SD))) :
            Grade="D"
            Grade_points=6
        elif(Total_marks>(Class_avg-(2.0*SD)) and
(Total_marks<=(Class_avg-(1.50*SD))) :
            Grade="E"
            Grade_points=5
        else:
            Grade="F"
            Grade_points=0
        list_courses+=[Course_credit,Grade,Grade_points]
    else:
        print("Invalid Input, Enter the input with great
care!")
        break
    else:
        TEE=float(input("Enter your Term-end marks of this course
out of 100 :"))
        Internals=float(input("Enter your internal marks out of
10:"))
        Lab_assignment=float(input("Enter your Lab assignemnt

```

```

marks out of 60 :"))
    TEE=(TEE/100)*30
    Total_marks=TEE+Internals+Lab_assignment
    if(TEE <= 100 and Internals<=10 and Lab_assignment<=60):
        if(Total_marks>90):
            Grade="S"
            Grade_points=10
        if(Total_marks>80 and Total_marks<=90):
            Grade="A"
            Grade_points=9
        if(Total_marks>70 and Total_marks<=80):
            Grade="B"
            Grade_points=8
        if(Total_marks>60 and Total_marks<=70):
            Grade="C"
            Grade_points=7
        if(Total_marks>50 and Total_marks<=60):
            Grade="D"
            Grade_points=6
        if(Total_marks>40 and Total_marks<=50):
            Grade="E"
            Grade_points=5
        else:
            Grade="F"
            Grade_points=0
        list_courses+=[Course_credit,Grade,Grade_points]
    else:
        print("The inputs are invalid, please enter the inputs
correctly again")
        break
    print("The course credits along with their Grade and Grade points
are displayed below\n")
    print(list_courses)
    #Calculation of the semester's GPA
    for i in range (0,len(list_courses),+3):
        GPA=GPA+((list_courses[i]*list_courses[i+2])/Total_credits)
        GPA=round(GPA, 2)
    print("The GPA of the current semester is :",GPA,"\\n")
    return GPA,Total_credits
list_GPA=[]
list_Total_credits=[]
Total_credits_completed=0
for i in range(0,8):
    Choice=int(input("Enter 1 to calculate the Semester's GPA and
CGPA, Enter 2 to exit :\n"))
    if(Choice==1):
        CGPA_=0
        CGPA=0
        GPA_,Total_Credits=GPA_Calculator()

```

```
list_GPA.append(GPA_)
list_Total_credits.append(Total_Credits)
Total_credits_completed=Total_credits_completed+Total_Credits
for i in range(0,len(list_GPA)):
    CGPA_=CGPA_+(list_GPA[i]*list_Total_credits[i])
CGPA=CGPA_/Total_credits_completed
CGPA=round(CGPA, 2)
if(CGPA>10.0):
    print("You have made a mistake in entering the inputs
please do enter the inputs with care!")
    break
else:
    print("Your CGPA is :",CGPA)
else:
    print("Thank you for using VIT'ian CGPA Calculator")
    break
```