

GPA Calculation Program

The following program computes a semester GPA and new cumulative GPA for a given student. This program utilizes the following programming features:

- ▶ tuple assignment

The following illustrates an example execution of the program.

```
This program calculates semester and cumulative GPAs

Enter total number of earned credits: 30
Enter your current cumulative GPA: 3.25

Enter grade (hit Enter if done): A
Enter number of credits: 4
Enter grade (hit Enter if done): A
Enter number of credits: 3
Enter grade (hit Enter if done): B
Enter number of credits: 3
Enter grade (hit Enter if done): B
Enter number of credits: 3
Enter grade (hit Enter if done): A
Enter number of credits: 3
Enter grade (hit Enter if done):

Your semester GPA is 3.62
Your new cumulative GPA is 3.38
>>>
```

Task: In IDLE, open a new project and save it as `gpaCalc_yourLastName`. Copy the code from the sample on the next pages. Test and revise the program, as needed.

```
1 # Semester GPA Calculation
2
3 def convertGrade(grade):
4     if grade == 'F':
5         return 0
6     else:
7         return 4 - (ord(grade) - ord('A'))
8
9 def getGrades():
10    semester_info = []
11    more_grades = True
12    empty_str = ''
13
```

```

14     while more_grades:
15         course_grade = input('Enter grade (hit Enter if done): ')
16         while course_grade not in ('A','B','C','D','F',empty_str):
17             course_grade = input('Enter letter grade received: ')
18             if course_grade == empty_str:
19                 more_grades = False
20         else:
21             num_credits = int(input('Enter number of credits: '))
22             semester_info.append([num_credits, course_grade])
23
24 def calculateGPA(sem_grades_info, cumm_gpa_info):
25     sem_quality_pts = 0
26     sem_credits = 0
27     current_cumm_gpa, total_credits = cumm_gpa_info
28
29     for k in range(len(sem_grades_info)):
30         num_credits, letter_grade = sem_grades_info[k]
31
32         sem_quality_pts = sem_quality_pts + \
33             num_credits * convertGrade(letter_grade)
34
35         sem_credits = sem_credits + num_credits
36
37     sem_gpa = sem_quality_pts / sem_credits
38     new_cumm_gpa = (current_cumm_gpa * total_credits + sem_gpa * \
39         sem_credits) / (total_credits + sem_credits)
40
41     return (sem_gpa, new_cumm_gpa)
42
43 # ---- main
44
45 # program greeting
46 print('This program calculates new semester and cummulative GPAs\n')
47
48 # get current GPA info
49 total_credits = int(input('Enter total number of earned credits: '))
50 cumm_gpa = float(input('Enter your current cummulative GPA: '))
51 cumm_gpa_info = (cumm_gpa, total_credits)
52
53 # get current semester grade info
54 print()
55 semester_grades = getGrades()
56
57 # calculate semester gpa and new cummulative gpa
58 semester_gpa, cumm_gpa = calculateGPA(semester_grades, cumm_gpa_info)
59
60 # display semester gpa and new cummulative gpa
61 print('\nYour semester GPA is', format(semester_gpa, '.2f'))
62 print('Your new cummulative GPA is', format(cumm_gpa, '.2f'))

```

Notes:

The program begins with the display of the program greeting on line 49. Lines 49–50 get the number of earned credits (`total_credits`) and current cumulative GPA (`cumm_gpa`) from the user. These two variables are bundled into a tuple named `cumm_gpa_info` on line 51. Since they are always used together, bundling these variables allows them to be passed to functions as one parameter rather than as separate parameters.

Function `getGrades` is called on line 55, which gets the semester grades from the user and assigns it to variable `semester_grades`. The value returned by function `getGrades` is a list of sublists, in which each sublist contains the letter grade for a given course, and the associated number of credits,

```
[['A', 3], ['B', 4], ['A', 3], ['C', 3]]
```

On line 58, function `calculateGPA` is called with arguments `semester_grades` and `cumm_gpa_info`. The function returns a tuple containing the semester GPA and new cumulative GPA of the user. A tuple assignment is used to unpack the two values into variables `semester_gpa` and `cumm_gpa`. Finally, these values are displayed on lines 61 and 62.

Function `calculateGPA` is defined in lines 24–41 with parameters `sem_grades_info` and `cumm_gpa_info`. A GPA is calculated as the total quality points earned for a given set of courses, divided by the total number of credits the courses are worth. The number of *quality points* for a given course is defined as a course grade times the number of credits the course is worth. Thus, assuming a grade of A is worth 4 points, B worth 3 points, and grades of C, D, and F worth 2, 1 and 0 points, respectively, to calculate the semester GPA for a student receiving A's in two four-credit courses, B's in two three-credit courses, and a C in a one-credit course would be,

$$\begin{array}{ccccccccc} (4 * 4 + 4 * 4 + 3 * 3 + 3 * 3 + 2 * 1) / 15 = 3.47 \\ \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \\ A \quad \quad A \quad \quad B \quad \quad B \quad \quad C \end{array}$$

where 15 is the total number of credits of all courses.

Similarly, in order to calculate a new cumulative GPA, the total quality points of the current cumulative GPA plus the total quality points of the new semester GPA is divided by the total number of credits the student has earned to date. Thus, to calculate a new cumulative GPA for a current cumulative GPA of 3.25 earning thirty credits, and a new semester GPA as given above (3.47 earning fifteen credits) would be,

$$(3.25 * 30 + 3.47 * 15) / 45 = 3.32$$

with 45 total earned credits. Thus, in function `calculateGPA`, local variables `sem_quality_pts` and `sem_credits` are initialized to zero. Their values for the courses provided in parameter `sem_grades_info` are computed in the for loop on lines 29–35. This loop also calculates the semester quality points and the number of credits of the current semester, assigned to local variables `sem_quality_pts` and `sem_credits`, respectively (at lines 32 and 35). Note that in the calculation of the semester quality points, function `convertGrade` is called to convert each letter grade to its corresponding numerical value. Finally, at the end of function `calculateGPA`, local variable `sem_gpa` is assigned to the total semester quality points divided by the total semester credits. Similarly, local variable `new_cumm_gpa` is assigned to the total quality points to date (`current_cumm_gpa * total_credits + sem_gpa * sem_credits`) divided by the total number of credits earned to date (`num_credits + sem_credits`). Finally, on line 41, a tuple is returned containing both of these computed values.

The remaining functions defined in this program are `convertGrade` and `getGrades`.

Function `convertGrade` is passed a letter grade, and returns the corresponding numerical value. Since the ordinal value (via the `ord` function) of letters in Python are sequential integers, determining the difference between the ordinal value of A and the ordinal value of a given letter grade allows the numerical value of the letter grade to be determined. For example, for a letter grade of A through D, its numerical value is determined and returned as,

```
return 4 - (ord('A') - ord('A')) → return 4
return 4 - (ord('B') - ord('A')) → return 3
return 4 - (ord('C') - ord('A')) → return 2
return 4 - (ord('D') - ord('A')) → return 1
```

Since there is no letter of grade E used, a grade of F has to be handled separately.

Finally, function `getGrades` returns a list of sublists of grades and credits entered by the user, as mentioned above. Thus, local variable `semester_info` is initialized to an empty list on line 10. The while loop at line 14 iterates until Boolean variable `more_grades` is `False`, initialized to `True` in line 11. The loop continues to iterate and append another pair of grade/credits to the list until the user hits the Enter key when prompted for a course grade (line 15).