Group project 2327

About

This is a project of the university of St. Gallen of the course programming – introduction level. The goal of the project was to create a program which calculates the GPA. Thanks to the program the user is able to enter for each semester his marks. The program calculates then the GPA. As an extra the program plots a Graph, which shows the development over the past semesters.

The program is inspired by the fact that at the university of St. Gallen not every mark has the same weighting. This leads to the problem, that if you want to calculate your GPA, it is not possible to use the commune formula: sum of marks divided by the number of marks. The fact, that for the calculation of the GPA it is necessary to take the value into account, makes the calculation rather cumbrous. Thanks to the GPA-calculator the user is able to enter his marks and its value. The calculator will provide a GPA, which has taken the value into consideration.

Pre-requests

The program works with Python 3.8. In order to run the GPA-calculator the libraries “numpy” “math” and “matplotlib.pyplot” have to be installed.

Instructions:

1. Start the gpa-calculator.py
2. Enter “yes”, “show” or “exit”. It does not matter if the input is written in uppercase or in lowercase letters.
3. If entered “yes”, insert “yes” or “no”. It does not matter if the input is written in uppercase or in lowercase letters.
4. If entered “yes”, insert in which semester you are in, followed be the input of how many marks you want to insert. Both inputs must be an integer.
5. If entered “no”, insert how many marks you want to insert. The input must be an integer.
6. If entered “yes” at step three The program asks you to insert in a row the name of the subject, the value of the subject and the mark.
    a. It is important that after each input (after the name, after the value and after the mark) to press enter.
    b. After entered the first subject, you can start inserting straightaway the next subject.
    c. The value and the mark must be an integer.
    d. “Value of the subject” is the weighting of the subject. Normally it is the number of ECTS.

Assignment specifications

1. Start the gpa-calculator.py
2. The user will have the possibility to choose between three options. He can decide if he wants to insert his marks. If he just wants to check his current GPA and his GPA development over the previous semesters, he can do so by choosing the second option. The third option is an exit button, which ends the program immediately.
3. First option: “Enter marks”:
   Sometimes you are just wondering, how a certain mark would have changed your GPA. In this case you want to calculate various options and check the outcomes. What you don’t want, is to change your current GPA. The program takes this into account and offers the user two further options. The first would be to enter the marks and safe them in a text file. If the user decides to check different possible outcomes, as already described, he needs to choose the second option.
   In the first option the user must enter two more inputs before he can insert his marks. The first input asks in which semester he is. This information will be used later for the file handling and for the construction of the graph. The second input wants the information of how many marks the user wants to insert.
   In the second option the program asks just for inserting the number of marks.

   The code in the background is for both options basically the same. This allows to explain the code just once.

   By knowing how many marks the user will input and thanks to the fact that there will be for each mark three inputs, it is possible to create a matrix. In the matrix the program will save the inputs from the user. For each mark it creates a row and each row has three columns (name of the subject, value of the subject, mark).

   The matrix looks like this:

Name of the subject 1	Value 1	Mark 1
Name of the subject 2	Value 2	Mark 2
Name of the subject 3	Value 3	Mark 3
…	..	…
…	..	..
Nam of the subject *n*	Value *n*	Mark *n*

   The user inserts then in a row *name of the subject, value of the subject* and *mark.* The inputs are separated by enter. This information is stored in the matrix above.

   As the user has entered all the marks, the program has everything to calculate the GPA. To do so, it has to extract the values from one to *n* and the marks from one to *n* and store the information into a second matrix, which looks like this:

Value 1	Mark 1
Value 2	Mark 2
Value 3	Mark 3
..	…
..	..
Value *n*	Mark *n*

Thanks to this format, it is possible to use numpy for the calculation.

The formula for the GPA is as followed.

1) Multiply each mark with its value.
   Code: result1 = np.multiply(convertcol2, convertcol3)
2) Calculate the sum of the results from step one.
   Code: result11 = math.fsum(result1)
3) Calculate the sum of the values 1 to *n.*
   Code: result2 = sum(map(float,convertcol2))
4) To get the GPA divide the result from step one by the result from step two.
   Code: GPA = result11/result2

The program will print then the GPA for the current semester.

3.1. Saving the data into a file (This step is just needed for the option: Entering marks and saving GPA):

To save the GPA, it is necessary to work with the file handling. In order to do so, it is required to open a file and write the required data (Semester: *n* and GPA of the semester *n*) into it. As at step three explained, the information Semester: *n,* was received from the user-input. It is possible to append the file with this information. To store the GPA in the file, the program just appends the result from step three.

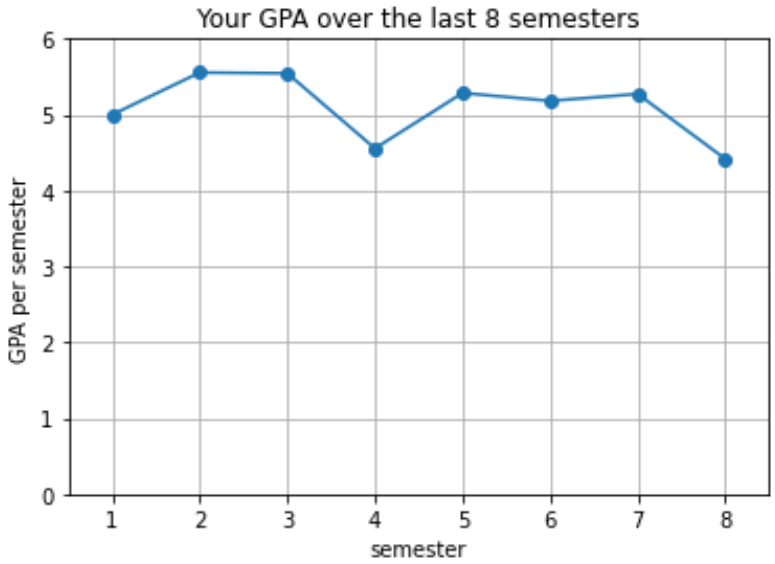
The file will look like this:

```
1  Semester:1
2  GPA:
3  5.0
4
5  Semester:2
6  GPA:
7  5.555555555555555
8
9  Semester:3
10 GPA:
11 5.545454545454546
```

3.2. Calculating the overall GPA

Up to this point the user just knows how good he has performed in the current semester. Thanks to saving all previous GPAs into the file, it is possible to calculate the overall GPA as well. To do so the program opens the file and takes out the third line of the text file (In the example above: 5.0) and saves it into an array. To receive the other GPAs, the program saves every fourth line from the text file into the array. Once done that, the program can calculate the overall GPA. The overall GPA is calculated by taking the sum of the array and dividing it by the number of elements in the array.

3.3. Plotting GPA-Development Graph (This step is just needed for the option: Entering marks and saving GPA):
The last step is to create a graph, which shows the development of the GPAs. To create the graph the program uses mathplot. The x-axis consists of the number of semesters (1 to *n*) and the y-axis consists of the GPAs from semester 1 to *n*. The label of the x-axis is defined as "semesters" and the label of the y-axis is defined as "GPA per semester". A possible plot could look like this:



4. Second option: "show":
With the option "show" the program gives the user as output the GPAs from Semester one to *n*, the overall GPA and shows him the GPA-Development Graph.
To come to this result, the first thing is to open the file, in which all GPA are stored. Once the file is opened, the program reads the whole file and prints it in the console. The code to get the overall GPA and the GPA-Development Graph, is the same as at step 3.2. and at step 3.3.
5. Third option: "exit":
With this option the program is ended and prints "Thanks for using the GPA-calculator" and "Goodbye".

Error handling

To ensure that the program works smoothly, I had to take the possibility of a false input into account.
The idea behind the error handling is as followed: The program converts the user-input into its Unicode characters and checks if the input is in the range of the should be Unicode characters. All this takes place in a while loop. So if the user-input is in the required range the user can move forward and exits the loop. If the input is not in the range, the user remains in the loop and can insert his input again.

I have four different user-inputs, which I have to take into consideration.
The first is the input, which gives the user the possibility to decide whether he wants to insert marks, show the GPAs from the previous semesters, or exit the program.

For the first input I have constructed the error handling as followed:

1. Create a while loop
2. Ask the user to input “Yes”, “show” or “exit”
3. Convert the user-input into its Unicode character.
   First the program has to split the user-input into strings with a length of one. Then it converts the strings into its Unicode character and stores it into an array.
4. Create an array, which contains the Unicode character A to Z and a to z. If all the elements of the input are in the range of A to Z and a to z, the program exits the loop. If the input is not in the required range the user as to enter his input again.
5. If the input is in the correct range, the program will check whether the input was “Yes”, “show” “exit” or neither of those.
   To be user-friendly, it should not matter whether the input is written in uppercase letters, in lowercase letters or a combination of those two. To take this into account the program converts the user-input into lowercase letters, so it does not matter if the user-input was originally a “Yes”, a “YEs" or a different combination, in the end every input is converted into an “yes”.
   Now the program checks if the user-input was “yes”, “show”, “exit” or neither of those. If the input was false, the user has to insert the input again. He remains in the while loop. Otherwise, the user moves forward.

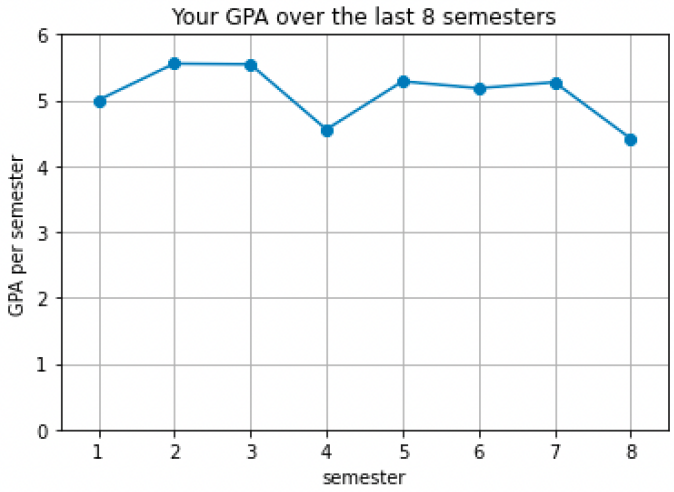
The second input consists of the question if the user wants to enter marks and save the GPA or if he just wants to enter some marks. The structure of the code is exactly the same as for the first input, the only difference is that this time it tests for “yes” or “no”.

The third input asks the user in which semester he is. And the fourth input asks the user how many marks he wants to insert. In both the user has to insert an integer. The code structure is similar to the one for the first input. The difference is that the program tests if the user-input is in the range of the Unicode character of 1 to 9. Because the program uses this code multiple time, it made sense to create an error testing function for each input.

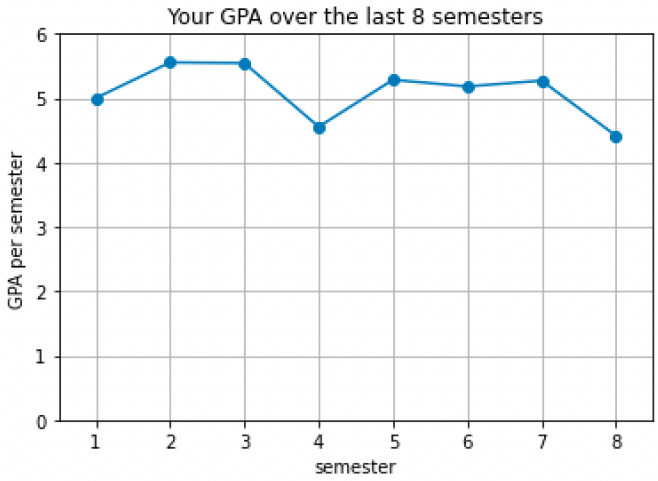
The last error handling is different than the previous four. Here the program uses try and except. Before the program asks the user in which semester he is, the program shows in which semester he was the last time when he used the program. The problem which occurs, is in the situation, in which the user is in the first semester. In this case the program has no data to access, and it stops with an error. With try and except it possible to solve this problem. In try: the program tries to access the data, if there is data to access the program takes out the required information. If the user is in the first semester the program stops and goes into except and prints the following sentence out: “You are in the first semester”.

Sample Output:

```
Welcome to the GPA-calculator

Please enter: (Yes) for inserting marks, (show) for show previous GPA, (exit) for ending the program: yes

For calculate and safe your GPA type (Yes), for only calculating type (No): YES


Your last semester was semester:7


In which semester are you in: 8

Enter the number of marks you want to insert: 3
Enter in rows for each subject: name of the subject, value of the subject, mark (separated by enter)

economics

4

4

law

5

5

mathematics

3

4


Your current GPA is 4.416666666666667

Your overall GPA is: 5.101686507936507
```



```
Welcome to the GPA-calculator

Please enter: (Yes) for inserting marks, (show) for show previous GPA, (exit) for ending the program: shOw


Semester:1
GPA:
5.0

Semester:2
GPA:
5.555555555555555

Semester:3
GPA:
5.545454545454546

Semester:4
GPA:
4.555555555555555

Semester:5
GPA:
5.285714285714286

semester:6
GPA:
5.181818181818182

semester:7
GPA:
5.2727272727272725

semester:8
GPA:
4.416666666666667


Your overall GPA is: 5.101686507936507
```




```
Welcome to the GPA-calculator

Please enter: (Yes) for inserting marks, (show) for show previous GPA, (exit) for ending the program: EXIT


Thank you for using the GPA-calculator
Goodbye
```