Grade Calculator

psp-05-05

1 Overview

In this lab we will be building a grade calculator to help you determine your current grade using a weighted average. You will create lists representing your score, and the max possible scores, for each assignment, quiz, or test, and create functions to caculate you score in each category, and to combine them using weights for each category. After this lab is complete, you could calculate your own class grade using this program structure.

2 Learning Outcomes

By the end of this project students should be able to:

- read and write programs that define simple functions;
- utilize functions with parameters and return statements;
- read and write programs with if-elif-else statements;
- work effectively with a partner using pair-programming;
- write an effective report that describes the students' problem solving process.

3 Pre-Lab Instructions

Do this part before you come to lab:

- Read Problem Space Chapter 5: Functions.
- If you need a quick review here is a link on Python function calls: http://www.tutorialspoint.com/python/python_functions.htm
- Sketch out pseudo code for a function that prints "Success" if passed True, and "Try again" if passed False. Also sketch how you would call the function you defined. Be prepared to show your pseudo-code to the lab aide at the beginning of lab.

- Read Problem Space 6.1: Lists
- Write down the python code you would use to access the first element in a list, and then a line to access the last item in a list.

4 Lab Instructions

Do this part in lab:
Step 1:
First, we need to set up our varaibles to be used in our program. This should be done in a function called main(). (The main function does not need take in any parameters nor return any values; it's purpose to initialize variables and make calls to other functions.) For each category, set up both a list with the maximum number of points possible for each assignment, and a list with the points earned for assignment in the category, as shown below.

Homework: 39/40, 40/40, 29/40, 40/40, 0/40, 5/5
Quizzes: 10/10, 10/10, 9/10, 2/10, 10/10, 10/10, 10/10
Test: 293/300, 284/300, 300/300

You should also set up a variable for each category that represents the total weight of that category. For example, homework may be set to 20%, quizzes to 20% and tests to 60%.

Add some temporary print statements here to ensure your list variables are being stored correctly.

Step 2:
Now we will create functions to caculate the average for each category and sum these together to come up with a letter grade.

Function: average(score_list, max_list)

Parameters:
score_list - A list of points earned.
max_list - A list of maximum possible points. Should be the same length as score_list.
Return:
returns a floating point number represnenting the average grade for a category.

Function: letter_grade(percent)

Parameters:

percent - a number between 0 and 1, representing your percentage score.
Return:
The string “A”, “B”, “C”, “D”, or “F” depending on the percentage, at the 90%, 80%, 70%, 60% thresholds.

Use these functions with the data created in step 1, and add print statements to ensure your functions work correctly.

Step 3:
Finally we will add a function to create a weighted average. This will be very similar to the average function, but will also multipy in our weights.

Function: average_weighted(score_list, max_list, weight)

Parameters:
score_list - A list of points earned.
max_list - A list of maximum possible points. Should be the same length as score_list.
weight - A number between 0 and 1 representing the weight of this category.
Return:
returns a floating point number represnenting the weighted average.

Once this is complete, you can use it on each category to generate your weighted averages. Add those together for your total grade.

Print the averages and grade for each category, and the final weighted average and grade as well. Round the averages to the nearest percent. Using the numbers from step 1 your output should resemble the following while using the grade values provided:

```
Homework grade: 74 (C)
Quiz grade: 87 (B)
Test grade: 97 (A)
Final Score: 90 (A)
```

When you have completed the lab run pep8 against your code until all formatting errors have been corrected and your code is PEP 8 compliant. See the Getting Started lab if you need instructions on running the program, or the pep8 documentation found here.

Extra Credit

In this lab you have the option of meeting additional requirements to earn extra points. Extra credit assignments can be done by a single individual if both parterns are not interested in completing the extra credit requirements. If you chose to complete the extra credit on your own, note that you have done so in

the comments section of the BBLearn submission.

The requirements for extra credit are to create a median function for your grade calculator without using loops, mapping functions or list comprehension. You will create two functions, a to_percent function, and a median function.

Function: to_percent(score_list, max_list)

Parametes:
score_list - A list of points earned.
max_list - A list of maximum possible points. Should be the same length as score_list.
Return:
A new list the same length as the input lists. Each value is the score's percentage of the max for that particular index. For example, a score list of [0, 10 ,10] with a max list of [10, 10, 30] should return a result of [0, 1.0, 0.5].

It is required that you complete this using only basic conditional (if) statments and division. You may use function recursion and list slicing. You may not use the for keyword, any mapping functions, or import from math to implement this function.

Function: median(score_list, max_list)

Parameters:
score_list - A list of points earned.
max_list - A list of maximum possible points. Should be the same length as score_list.
Return:
The median percentage score of the inputs. For example, a score list of [0, 10, 10] with a max list of [10, 10, 30] should return a median score of 0.33.

To acomplish this you may not import anything from math. You can use both the list .sort() method, as well as using the to_percent function you wrote previously.

Once you have implemented these functions, use the median function you wrote to show the median for homework, quizes and tests. Then submit your code along with your lab report.