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 will set up our varaibles to be used in our script. For each of these categories, set up both a list with the number of points for each assignments, and a list with the maximum points possible for this category.

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 for the total weight of that category. Homework should be set to 20% quizzes to 20% and tests to 60%, such that all three add up to one.

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

Step 2:
Now we will create functions to caculate the average for each section, and to come up with a letter grade.

Function: percentage_per_category(score_list, max_list)

"""Calculates the percentage of possible points earned in a category."""
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 representing the percentage of points earned in the given category.

Function: letter_grade(percent)

"""Return the letter grade equivalent of the given percentage.
90%+ = A
80%-89% = B
70%-79% = C

60%-69% = D
59% or lower = F
"""
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 give us a category’s portion of the final weighted score. This will consist of us multiplying the category’s percentage from the percentage_per_category() function by the category’s weight.

Function: weighted_score(percentage, weight)

"""Calculates the weighted contribution of a category
to the overall course grade.
"""
Parameters:
  percentage - The percentage of possible points earned in this category
  weight - A number between 0 and 1 representing the weight of this category.
Return:
  returns a floating point number representing the amount the given category contributes to the overall grade .

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

Print the percentages and letter grade for each category, and the final weighted score and letter 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 0.5.

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.

5 Lab Report

Each pair of students will write a single lab report together and each student will turn in that same lab report on BBLearn. Submissions from each student on a pair should be identical.

Your lab report should begin with a preamble that contains:

- The lab assignment number and name
- Your name(s)
- The date
- The lab section

It should then be followed by four numbered sections:

1. Problem Statement

In this section you should describe the problem in **your** own words. The problem statement should answer questions like:

- What are the important features of the problem?
- What are the problem requirements?

This section should also include a reasonably complete list of requirements in the assignment. Following your description of the problem, include a bulleted list of specific features to implement. If there are any specific funtions, classes or numeric requirements given to you, they should be represented in this bulleted list.

2. Planning

In the second section you should describe what planning you did in order to solve the problem. You should include planning artifacts like sketches, diagrams, or pseudocode you may have used. You should also describe your planning process. List the specific data structures or techniques you plan on using, and why.

3. Implementation and Testing

In the third section you should describe how you implemented your plan. As directed by the lab instructor you should (as appropriate) include:

- a copy of your source code (Submitted in BBLearn as a .py file)
- a screen shot of your running application / solution
- results from testing

4. Reflection

In the last section you should reflect on the project. Consider different things you could have done to make your solution better. This might include code organization improvements, design improvements, etc.

You should also ask yourself what were the key insights or features of your solution? Were there alternative approaches or techniques you could have employed? How would these alternatives have impacted a different solution?

5. Partner Rating

Every assignment you are required to rate your partner with a score -1, 0 or 1. This should be submitted in the comment section of the BBLearn submission, and not in the report document. You do not have to tell your partner the rating you assign them. A rating of 1 indicates that your partner was particularly helpful or contributed exceptional effort. A rating of 0 indcates that your partner met the class expectations of them. Rating your partner at -1 means that they refused contribute to the project, failed to put in a resonable effort or actively blocked you from participating. If a student recieves three ratings of -1 they must attend a mandatory meeting with the instructor to dicuss the situation, and recieving additional -1 ratings beyond that, the student risks losing a letter grade, or even failing the course.

Colophon

This project was developed by Dr. James Dean Palmer of Northern Arizona University. Except as otherwise noted, the content of this document is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.