Assignment #6

(Design a Larger Python Program)

Step 1: Problem Analysis.

a. The programmer can only create the program with functions beside the call on the main function at the end of the program. There are required functions for the program, which are the main function, get_initial_input function, get_scores function, calculate_weighted_avg function, and the calculate_class_grade function; but the programmer can create additional function if wanted. The task at hand is to create a program that will calculate the final grade of a certain class by information of a variety of scores of different categories of the class inputted by the user.
b. The programmer would have to know the usage and creation of defining a function, how to create list, how to use loops, and how to involve all the functions into the main function overall.

Step 2: Program Design.

1. Define the function get_initial_input.
2. Within that function, prompt the user for the number of tests/assignments/quizzes/labs as an integer as a return.
3. Define the function get_scores.
4. Within that function, prompt the user for the scores of the number of tests/assignments/quizzes/labs as floats as a list.
5. Define the function calculate_weighted_avg.
6. Within that function, prompt the user for each of the weighted percentages for each of the tests/assignments/quizzes/labs as floats.
7. Take those weighted percentages and multiple them to its corresponding category scores from the get_scores function.
8. Define the function calculate_class_grade.
9. Within that function, the final calculations from the calculate_weighted_avg function will be summed up as a float.
10. Define the main function.
11. Within that function, call the function get_initial_input.
12. Call the function get_scores.
13. Call the function calculate_weighted_avg, except for the labs category where the scores are needed to be summed up and then multiplied by the inputted weighted percentage.
14. Call the function calculate_class_grade.
15. Have all the code that just has been created within this main function looped.
16. Prompt the user if they would like to calculate another grade for the class. If so, have the loop go again, and if not, break the loop.
17. Outside all of the created functions call the main function at the end of the program.

Step 4: Program Testing:

Average Case(s):

- When the values are able to be correctly read by the user, the program will compute all the calculation and print all of the calculated outputs of the program to be presented to the user.

Extreme Case(s):

- When an invalid value is inputted into the program, it'll become an error in the program.
- Another case would be if the values of the weighted percentages do not add up to 100, the final grade will be skewed.

Error Handling:

What are the errors that can occur in your input with your current design?

It will not be able to handle floats if the user is only able to input integers into the program.

What other functions will you need to handle errors in the input?

Possibly just create an if else command within the function(s) where that error could occur.

What functions will call these error handling functions? Design these functions and the picture of how these functions will interact with other functions.