Assignment #6
Design a Larger Python Program
Due: Monday, 11/17/14, 11:59pm

In this assignment, some of you will design your first big programming project. You will design the flowchart and/or pseudocode for a program that calculates your numeric grade in a class based on the following inputs and processes:

1. Ask the user for the number of tests, assignments, quizzes, and labs in their course.
2. Ask the user if there is a final with a separate weight from the tests above, e.g. a course could have 2 tests, each weighing 12.5%, and 1 final weighing 15%.
3. For each category having a number > 0
   a. Prompt the user for the weighted percent, out of 100%, which should total 100% for all categories!!!
   b. Get the score(s) for the category.
   c. If the category is labs, then sum all the scores
   d. Else, average the scores.
   e. Calculate the weighted average for the category.
4. Using the weighted average of each category, calculate the grade in the course.
5. Ask the user if he/she wants to calculate a grade for another class.
6. If the user responds yes, then go back to step 1.
7. Else, end the program.

Requirements, you must have a main function, i.e. def main():, which creates any variables you need for the program and calls other functions using variables created inside of the main function. The only call or variable creation you are allowed to have outside of main (or any other function) is the call to main(), at the very bottom of your program. In addition to your main function, you must define these other functions:

   get_initial_input()
   get_scores()
   calculate_weighted_avg()
   calculate_class_grade()

You can define other functions as you see fit, but at minimum you must have these listed above.

At this time, you are required to draw a picture of the functions and which functions are called from that function, starting with main. In addition, write the function headers and pseudocode for each of the functions above, including main. In these function headers, you must describe the functionality of the function, the parameters, return values, pre-conditions, and post-conditions.
http://classes.engr.oregonstate.edu/eecs/fall2014/cs160-001/160_style_guideline.pdf
The pre-conditions are anything that you can assume true about the parameters, at the time when the function is called. For example, a pre-condition for `calculate_weighted_avg()` might be that the scores are valid floating point numbers, which means this function would not have to catch errors associated with bad input. The post-conditions are anything that you can assume true about the arguments after the function executes.

(90 pts) Begin by designing your program using these steps, and write steps 1, 2, and 4 on paper or in a text editor. At this time, we are only designing but not implementing anything!!!

- **Step 1: Problem Analysis.** (15 pts)
  a. Comments about the problem to aid in understanding it.
  b. Description of the knowledge base (this list would include what you would be expected to know to follow the solution).

- **Step 2: Program Design.** (50 pts) First draw a picture of how each function will interact with other functions, i.e. draw boxes to represent the function and draw lines between the functions that call other functions. Then, list the specific steps needed in each of the functions, including the main function. This also includes the function headers that provide information about the parameters and return values. Remember, you have to be very explicit here to make sure the computer can accomplish the task using your directions.
  1.
  2.
  3.
  ....

- **Step 4: Program Testing.** (25 pts)
  Create a Test Plan with several test cases including the average and extreme cases.

**Error Handling (10 pts):**
What are the errors that can occur in your input with your current design? What other functions will you need to handle errors in the input? What functions will call these error handling functions? Design these functions and the picture of how these functions will interact with other functions.

Electronically submit your design as a pdf by the assignment due date, using TEACH: [https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth](https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth)