Assignment #6
Design a Larger Python Program
Due: Sunday, 10/29/17, 11:59pm

In this assignment, some of you will design and begin to implement your first big programming project. First, 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, exercises, 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. If the category is labs, then prompt for each score and sum all the scores.
    c. Else, ask the user if the scores are out of varying points or out of a constant number of points.
        - If the scores are out of varying points, then get each score and number of points each score is out of to determine the percentage.
        - Else, get the constant number of points for all scores and get each score to determine the percentage.
        - Determine the percentage average for the category.
    d. 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.

Please see an example of the three following steps for a subset of this problem: Polya_template.pdf

(75 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!!! Take this seriously, and write every single step needed to complete the task above.

- **Step 1: Problem Analysis**. (10 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**. (40 pts) You can draw a flowchart or pseudocode for your design, but you need to be thorough. Your design needs to make sure to include error checking for all invalid input.

- **Step 4: Program Testing**. (25 pts)
  Create a Test Plan with several test cases including the bad, average and extreme/edge cases. Come up with enough that you can use to convince yourself that the program works properly.

(25 pts) Step 3: Implementation
Try your best to implement your design from above. You will be graded on effort, rather than correctness. Try to implement a little bit of your design at one time, making sure the functionality for that piece is correct before moving to the next step or piece of your design.

Electronically submit your **design** as a pdf and **source code** as a .py file by the assignment due date, using TEACH:
https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth