Assignment #3
Conditionals, Looping, Begin Functions
Due: Sunday, 2/05/17, 11:59pm

(90 pts) **Problem Statement:** Write a C++ program that calculates averages for specific categories in the class, as well as predicts your grade depending on future grades.

This assignment will **contain at least 3 functions**, **loops, and decisions**. In this assignment, **we are going to make a grade calculator!!!** You will continually **ask the user for which category he/she wants to calculate an average: tests, assignments, labs, recitations, or overall for the class, as well as quit**. Based on the input, you will **prompt the user for the scores for that category** to calculate the average.

Scores/grades can be floating point numbers, and categories have different point scales, which you will get as input from the user. These point values may vary or be uniform, which you also need to get from the user. If the point values vary, then you need to get a point value for each score. If they are uniform, then you can enter one point value for all scores.

Lab, assignment, and test averages are all calculated the same based on the number given, point values, and scores. However, the recitation average differs because it has subcategories. You must ask the user for the subcategory information for quizzes, designs, and critiques, including the weight for each. If a weight is zero, then you can assume the user doesn't want to enter/include grades for that subcategory.

You must initialize all category averages to zero, and you will prompt your user for the weights of the different categories when they choose to calculate the class average. If a weight of zero is provided, you do not include this in the calculation.

Example Run:

```
Would you like to calculate an average for 1) labs, 2)
assignments, 3) recitations, 4) tests, 5) the class, or 0)
to quit? 1

How many labs? 4
Do the point values vary (0-no, 1-yes): 0
Out of how many points? 10

Lab 1 score: 10
Lab 2 score: 10
Lab 3 score: 10
Lab 4 score: 10

Your lab average is 100
```

```
Would you like to calculate an average for 1) labs, 2)
assignments, 3) recitations, 4) tests, 5) the class, or 0)
to quit? 3

Quiz weight: 40
Design weight: 0
Critique weight: 0

How many quizzes? 2
Do the point values vary (0-no, 1-yes): 1

Quiz 1 points: 10
Quiz 1 score: 10
Quiz 2 points: 9
Quiz 2 score: 9

Your recitation quiz average is 100

Your recitation average is 100

Would you like to calculate an average for 1) labs, 2)
assignments, 3) recitations, 4) tests, 5) the class, or 0)
to quit? 5

Test weight: 0
Recitation weight: 20
Lab weight: 10
Assignment weight: 0

Your percentage in this class is: 100

Would you like to calculate an average for 1) labs, 2)
assignments, 3) recitations, 4) tests, 5) the class, or 0)
to quit? 0
```

For example, here are some functions you might want to have:

- start_calculator();
- is_positive_int();
- is_positive_float();
- is_in_range();
- calculate_lab_average();
- calculate_recitation_average ();

- In addition, create any other functions you might need to properly modularize your code

(10 pts) **Extra Credit:**
Each function, including main, has a maximum of 20 lines of code (comments and blank spaces do not count).

(10 pts) **Program Style/Comments**
In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example header to include. Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don't align everything on the left or put everything on one line!
http://classes.engr.oregonstate.edu/eecs/winter2017/cs161-001/161_style_guideline.pdf

You are graded on having a **header, function headers with pre/post conditions, proper comments, and readable code with indentation** and vertical spacing that is CONSISTENT throughout your program. DO NOT align your entire program on the left side. This will cause you to automatically lose the full 10 points. In addition, do not forget your program header!!!

Electronically submit your C++ program (**.cpp** file, not your executable!!!) by the assignment due date, using **TEACH**.