Chapter 5 Lab : Functions

Goal: This lab is intended to help you understand functions, scope, passing, and returning. You should hand in at least one *.py* file for each of the below problems. Please name and number the problems appropriately (something like *Lab5_1_YourName.py*). Note that you will also have to include any modules developed as part of the problems (if you choose to do so). Each code should contain extensive comments in the style discussed in class and output should be clearly displayed and labeled. Make sure that you run the test cases as specified, and do *not* have user input unless indicated (only #4 should have input from the user).

Structure your codes as you were shown in the lectures. Put the functions above the main, and do not mix-and-match portions of code (have "regular" main code in between functions). For example:

```
def solveLinear(m,b):
    return -b/m;
print("The solution to",2,"x +",3,"is x=",solveLinear(2,3))
```

1. Create two functions, *mileToKm* and *kmToMile* which convert from miles to kilometers and back, respectively. Use at least three decimal places for each conversion factor. Each should accept a number as input and return a number. In the *main* test your code on 1 *mile* and 1 *km*. Then test to make sure each function inverts the other function (under composition) using a random integer distance between 5 and 15 (miles for one case, kms for the other). In other words, make sure, in math terms:
   $x = mileToKm(kmToMile(x))$ and $x = kmToMile(mileToKm(x))$
   Print out what x is chosen, then print the result of the composition operation. Do not use logical operations to see if the inversion operations result in equal values, as computer round-off will result in them to be off a tiny bit. Example output:
   1 mile is 1.60934 km

2. Create a function, *nflQB* which takes in the following QB passing data

   - Completions, C
   - Attempts, A,
   - Yards, Y
   - Touchdowns, T
   - Interceptions, I

   The function should return a single number which corresponding to the NFL Quarterback rating. The rating can be found, in most cases, by using the following formula:
   $$\frac{25}{12}\frac{40C + A + 2Y + 160T - 200I}{A}$$
   Some exceptions exist for abnormally high or low numbers (which we will ignore here). In your main code, test (and output results) on both Dak Prescott and Cam Newton for the entire 2018 season (find the stats here: https://www.pro-football-reference.com/years/2018/passing.htm). Make sure you end up with the same passer rating as on the website. Example output:
   Patrick Mahomes QB Efficiency Rating: 113.8

3. Make a function, called *calcWeight* which takes in two arguments. The first is the mass of an object. The second is the units of the mass (either "g" or "kg"). Your code should output the weight in both pounds and Newtons (in that order). If the units are invalid, your code should print an error message to the terminal and return *math.nan* for both weights. In your main code, test by outputting results for 20 kg and 1337 g. Output only one decimal place. Example output:
   6347g mass, on earth, is equal to: 14.0 lbs or 62.2 N
   Error (invalid input) - 999p mass, on earth, is equal to: nan lbs or nan N

4. Create a code which administers a set of practice mathematics problems. You should use at least one function in addition to your main code.

- Each problem should draw two random positive one-digit numbers.
- Either a '+' or '*' should be chosen randomly as the operator.
- The practice problems should run as such
  - ⋆ Prompt with the operation. For example: 2 + 4 =? (or, say, 3 ∗ 5 =?).
  - ⋆ Read in the potential answer.
  - ⋆ Display a correct or incorrect message.
  - ⋆ If the student is incorrect, re-display the previous problem until they get the right answer.
  - ⋆ If the student is correct, provide them with a new question.
  - ⋆ Exit the code if a student enters 0 as an answer.

Note that you may structure your functions and code however you would like, but you *will* be graded on style. Repeated code blocks will result in points taken off. Similarly with excessive use of unnecessary functions. Example output:

Addition & multiplication practice! Enter 0 to quit.

2 + 2 = ?