LESSONS LEARNED

PS0
Purpose: confirm installation of Python, write a simple program, and submit it.

- import math
- call math functions
- convert int to string and string to int
- print to the screen and take input
- create and call a function

PS1
Purpose: intro to control flow and formulating computational solution to a problem.

- initialize state variables
- take multiple inputs
- perform multi-variable calculations
- create effective while loop and if-else conditions
- make use of '+=' symbol when adding to the same variable
- how to perform bisection search
- convert int to float and float to int
- use global variables, 'break' statement, and boolean values
- reset variables on each iteration
- use a counter (ie. mos)
- troubleshoot code and test in isolation to identify and rectify problem snippets

PS2
Purpose: intro to creating functions and using looping mechanisms.

- import String to access string functions
- 'for' loops, 'if' statements, and counters can be used in combination to iterate over all characters in a word
- initialize an empty string in order to track guesses
- there are different uses for return and print
- use string.ascii_lowercase to get all the letters in the alphabet
- variables and functions can be embedded in print statements
- tracking multiple counters (ie. guesses and warnings) and altering them based on different conditions (losing warnings first, 1 v. 2 guesses, etc.)
- verifying input (ie. isalpha(), asterisk, is a vowel, or is not)
- breaking reusable parts into functions
- making all input lowercase
- calling functions within other functions
- compiling a score for the game and sharing with the User if they win
- terminating the game
- removing whitespaces to ensure words have consistent lengths via .translate({ord(c): None for c in string.whitespace})
- using if statements, while loops, and boolean variables to track whether one word (with underscores) matches another word

- checking and sharing all words that match with another word
- tracking alphabetical letters that were guessed
- creating 2 versions of hangman
- calling on a text file with 55000+ words stored in it

PS3

Purpose: create a Word Game, similar to Scrabble or Words with Friends.

- how to create a shallow copy of a dictionary
- to iterate over keys and values in a dictionary
- use of .find() to return the index
- to replace characters in a word
- to return function calls as conditions, use of .random()
- to play multiple hands in a game
- to create and call multiple functions in order to build a game
- to follow pseudo code
- to test sections of code in isolation
- to write and then implement what was described in pseudocode

PS4

Purpose: take input string and use recursion to determine all possible re-orderings of characters.

- how to use recursion to 'enumerate': iterate over and store permutations of the remainder
- OOP (Object Oriented Programming): to model real world entities as software objects
- Classes: blueprint for how something should be defined
- to use getter, setter methods and '_init_' to initialize the class
- to return a dictionary of mapped letters
- when we've reached the end of the 26 letters of the alphabet, we can use the '%' (remainder function) to account for wrap-around cases.
- to populate a dictionary with key-value pairs
- code testing: adapt, use print statements, and observe effects in isolated sections of code
- how to encrypt / decrypt a message per Caesar cipher and random vowel cipher
- to use '.split()' to split a longer message into a list of words
- to use prior create functions (ie. 'get_permutations)

PS5

Purpose: use object-oriented programming to build a program to monitor an RSS feed.

- Import pytz as pytz as this older problem set (ca. 2016) didn't seem to import it correctly for my newer version of Python.
- To store info in an object, write a class, NewsStory
- Initialize constructor 'def __init__(self, …):' with proper arguments
- Create simple get methods for specified arguments.
- Define subclasses (of Trigger).
- Make use of 'maketrans' function to map characters. For example, to remove punctuation and spaces to more accurately process the words in a string.
- Use 'strptime()' to convert from string to datetime object and '.replace()' to update time zone.

- Make use of evaluate method to actually make what happen what we want to happen (ie. returning NOT the result of a trigger)
- To filter through stories for certain triggers.
- To use commas for delineation in a string (ie. arg = line.split(',')).
- To append items to a list with 'ADD' as the signal word.
- To create key-value pairs using different signal words.
- To return a list of triggers.
- To update the text file so that it searches for current events.
- For 'object display issue': my code was compiling and it matched similar examples that I'd found online yet it was not displaying due to an object issue. I'd spent some time trouble-shooting but ultimately realized that there's a chance this issue could be due to a version issue or because the supplemental code is old and thus I may end up chasing my tail for hours rather than learning. Thus, this code was good enough for me and given a class situation, if I couldn't have clarified this issue with other students / a TA / the prof, then I would have submitted it for time efficiency sake. Due to this I would not give this problem set 100%, rather 80%.