LAB #8
More Python Programming

1. **Pre-Lab**: Make sure you look over these designs we talked about in class for Problems #1-3 from the 11/09 and 11/12 lectures. This will ensure that you understand the programs that you are supposed to write.

   [Lecture Problems Pseudocode](#)

   **You are to work alone for this lab!!! We need to determine who is struggling, and the only way we can do this is to have you work individually!!!**

2. You will write the Python code to solve each of the three problems. There are two designs for Problem #1, and you are required to implement both designs!!! As we talked about in class, these designs do not handle erroneous input, and at this time, you do not need to handle errors. The pseudocode for these problems does not include prompts for the user. It is your job to determine an appropriate prompt for the user!!!

   As you code or after you code, you need to insert a comment above **EVERY** line of code explaining what each line of code does in your own words. If you can't explain what the line of code does, then ask a TA to help you!!! You insert Python comments using the #. Example:

   ```python
   #This prompts the user for a score, reads the score as a string from the user, 
   # turns it into a floating point number and assigns it to the variable score
   score = float(input("Please enter a score: "))
   ```

   **In order to receive 8 points for this part of the lab, show your Python code for each problem design with a comment explaining every line in your program to a lab TA!!!**

3. Now, add error handling to the first two problems.

   In Problem #1, what happens when a user enters a 0? What happens when a user enters a 1 in the first design versus the second design? What happens when the user enters a 2 in the first design versus the second design? What happens when the user enters a floating point number? What happens when the user enters characters instead of numbers? Change your code to handle these errors.
In Problem #2, what happens when the user enters a number less than or equal to 1? What happens when the user enters a 2? What happens when a user enters input that isn’t a number? Change your code to handle these errors.

Show a lab TA your error handling to receive 2 points on this lab.

Extended Learning

What if we wanted to retain the score values in Problem #1 and the deposits, checks, and fees in Problem #3? We would need to create a list to hold the values. You have already worked with one kind of list in Lab #7, which is a string. A string is a list of characters, and we accessed each character using the brackets, [], and the position in the list you want to access, starting at 0.

You can create a list in Python by typing `scores=[]`. Then you can add contents to the list by using `append`, i.e. `scores.append(89)`. You can access and view the contents of the list at a particular location by using the brackets, i.e. `print("The first score is:" + str(scores[0]))`.

Try rewriting Problem #1 and #3 using lists/1-d arrays