



ISAT 252—Analytical Methods IV

Programming and Problem Solving

Python Lab #7: Loops, Files and Exceptions

Due Date: Friday March 27, 2015

Objectives:

- Use looping to solve programming problems
- Know the differences between different types of looping
- Be able to identify and implement the most appropriate type of looping in various situations

Deliverables:

1. Soft copies of:
 - a. Your planning documents (pseudo-code or flowchart)
 - b. Your working program and source code
 - c. Your answers to the worksheet questions
2. Hard copies of:
 - a. Your planning documents
 - b. Your source code
 - c. Your answer to the worksheet questions

The Scenario: Friends of the Shenandoah River Revisited

You are a volunteer member of an organization called the Friends of the Shenandoah River, which monitors water-quality issues in the Shenandoah Valley. Water quality data is gathered weekly at a number of sites along the river by various volunteers and the data is then uploaded to their website (<http://fosr.org/data>). You have been asked to create a computer program that can read the data files that have been uploaded to the site, and then display statistical calculations based on that data.

You'd like to implement the following functionality:

1. You want to give the user a choice of one of two files, "Turbidity.txt" and "LabpH.txt" which you will find attached. Each file has in the first row the name of the item being measured and then an unknown number of readings. If you ask the user to enter the name of the file, be sure upper or lower case doesn't matter. Also, be sure you give a good message if the program can't find the file that the user requests.
2. You want to be able to read a file with any number of rows, but you won't know ahead of time how many rows of data will be in the file. You do know, however, that there will be no missing values, i.e. empty cells, in the data tables. (Hint: **See chapter 6 on how to read from the file until the end of the file is reached.** Which kind of loop will you need?) Your code may algorithmically look something like this:

```

# input name of file

# open the file

# get all input from file (NOTE: first row is data header)

while not the end of the file:

    #read a line of data and convert it numerical value

    #keep a running total

    #increment loop counter

# close file
file_object.close()

```

NOTE: You can also use a for-loop do the same thing. The choice is yours!

3. Your program should calculate the average value in the column that gets read in and then display the column title and average value in an easy-to-read format in your output.
4. Make your application as user-friendly as you can. Your program should **NOT** crash unexpectedly (*i.e. input validation and exception handling should be used*) and it should look professional.
5. Your program should present the output in SENTENCE form telling what was measured and the average reading.

HELPS on determining the variables needed:

1. What inputs will you have? What outputs are needed?
2. Inside the loop, you will need to keep a running total of the values as they are read in.
3. After the loop, you will need to divide by the number of items that were entered. That means you will also have to count them inside the loop as they come in.

Remember, **all** programmers, no matter what skill level, get stuck. In fact, it is the opportunity to get stuck, and then use your brain to figure out how to get unstuck that makes programming fun. Once again, here are some of the tried and true strategies for getting yourself out of a dead end:

- **Take a break**—you'd be surprised how many times a solution will come to you when you are doing something else.
- **Google it**—this is a great resource, especially if you use the word “tutorial” in the query for what you're trying to do.
- **Use your textbook**—the book is a great reference. You can frequently find short and good examples that show you how to do what you want. Almost all of the Python code you'll want to use can be found in the textbook.

ISAT 252—Analytical Methods IV—Programming and Problem Solving
Worksheet #7: Loops, Files and Exceptions

1. Write a set of nested loops that display 10 rows of # characters. There should be 15 # characters in each row.

2. Write code that opens an output file with the filename number_list.txt, but does not erase the file's contents if it already exists.

3. Write code that does the following: opens an output file with the filename number_list.txt, uses a loop to write the numbers 1 through 100 to the file, and then closes the file

4. What will the following code display?

try:

```
    x = float('abc123')
    print('The conversion is complete.')
except IOError:
    print('This code caused an IOError.')
except ValueError:
    print('This code caused a ValueError.')
print('The end.')
```