Laboratory VI

COMSC-122

Fall 2017

Example of How to Find Largest # in file

```
# Program 6-32  largest.py
# This program displays the largest Integer saved in a file.
def main():
    filename = input('Please Enter the Filename desired: ')
    infile = open(filename, 'r')
    largest = float(infile.readline())      # Start by assuming the first number is largest
    for line in infile:
        amount = float(line)
        if amount > largest:                # If subsequent # is larger, make that the largest
            largest = amount
    infile.close()
    print('The largest number in ', filename, ' is ', format(largest, ',.2f'))
    exit=input('')
main()
```

Program 6-29 (sales_report3.py)

```
1  # This program displays the total of the
2  # amounts in the sales_data.txt file.
3
4  def main():
5      # Initialize an accumulator.
6      total = 0.0
7
8      try:
9          # Open the sales_data.txt file.
10         infile = open('sales_data.txt', 'r')
11
12         # Read the values from the file and
13         # accumulate them.
14         for line in infile:
15             amount = float(line)
16             total += amount
17
18         # Close the file.
19         infile.close()
20
21         # Print the total.
22         print(format(total, ',.2f'))
23     except Exception as err:
24         print(err)
25
26 # Call the main function.
27 main()
```

Program 6-29

Printing all the applicable Default error messages which might have caused the error to be thrown, using the Exception clause.

Laboratory VI-A
Compute Statistics of a File of Integers

- You are asked to write a program that will generate the following statistics from a file of integers that you have been given called numbers.txt
- Your program will generate:
  - The value of the smallest number in the file.
  - The count of all the numbers in the file.
  - The sum of all the numbers in a file.
- Call your program *YourName*-LabVI-A.py
- Have your instructor look at your work so that you can receive proper credit.

LaboratoryVI-B

- Now add a **try / except** construct as shown in Program 6-29.
- Once you have that working try opening the following two files, and see what you get:
  - numbers1.txt
  - badfile.txt
- Call your improved program: *YourName*-LabVI-B.py
- Show the instructor how your program handles these two exceptions.

Generating Floating Point Random Numbers

- `random` function: returns a random float in the range of 0.0 and 1.0
  - Does not receive arguments
- **`uniform` function: returns a random float but allows user to specify range**
  - **Here's the syntax of its use [note its similarity to the randint() function]**
  - **random.uniform(arg1, arg2)**

Laboratory VI-C
Generate Random Floating Point #s

- If time permits, now create a program, similar to the one you generated in last week's lab, but instead of generating a file of random integers, it generates a file of random floating point numbers called float_numbers.txt.
- Like last week's file, the program should accept the three inputs:
  - The number of random numbers you want generated
  - The Maximum value of random number you want generated.
  - The minimum value of a random number you want generated.
- Call your program *YourName*-LabVI-C.py
- Run your program *YourName*-LabVI-B.py on this file of numbers and see what results you get.