More on File Input/Output

CS 8: Introduction to Computer Science, Winter 2019
Lecture #12

Ziad Matni, Ph.D.
Dept. of Computer Science, UCSB

Administrative

- **Hw06 DUE TODAY**
- **No Homework This Week!**

- **Lab06** – will be issued for Tuesday
  - Due by next week Monday

- You are still working on Project #1… right?

- **MIDTERM #2 is on Wednesday**

Midterm #2

- **What's going to be on it?**
  - Functions
  - Conditionals
  - Loops
  - String Formats
  - File I/O
  - ~~Random Numbers (and other Math stuff)~~

Lecture Outline

- File Input / Output

- Review for Midterm #2

File I/O: Simple Example

Example of READING from a file

```
infile = open('DataFile.txt', 'r')

line = infile.read()
# read everything in one string!

print(line)

infile.close()
# DON'T FORGET TO CLOSE!!!
```

Example of WRITING to a file

```
outfile = open('MyOuts.txt', 'w')

x = 3
y = 4
n = (x + y)**y

outfile.write('Number' + str(n))

outfile.close()
# DON'T FORGET TO CLOSE!!!
```

Different Ways of Reading File Input

```
line = infile.read()
                      # Read everything into 1 string
line = infile.read(n)
                      # Read the first n chars into 1 string
line = infile.readline()
                      # Read 1 line (ends in '\n') into 1 string
line = infile.readlines()
                      # Read all lines into 1 list
```

DEMO!
Let’s try it!

File I/O: More Examples

Example of READING from a file

```
filename = input
("What is the name of the file to
open? ")

InFile = open(filename, 'r')

count = 0
for line in InFile:
    count += 1
    print(line)
print("There are", count, "lines in
the file", filename)

InFile.close()
```

Example of WRITING to a file

```
filename = input
("What is the name of the file to
open? ")

OutFile = open(filename, 'w')

newl = '\n'
for n in range(10):
    OutFile.write('Number' + str(n)
+ '\n')

OutFile.close()
```

Read File

Example of READING from a file

```
filename = input
("What is the name of the file to
open? ")

InFile = open(filename, 'r')

count = 0
for line in InFile:
    count += 1
    print(line)
print("There are ", count, " lines
in the file ", filename)

InFile.close()
```

***open()** function, using the **'r'** option means that we want to READ this file. Note that **filename** is a string.*

*This is what we're doing to the lines that we read from the file. Note that the use of the **print()** function here means that the output goes to **"standard output"** (i.e. your screen)*

*Always **close()** the file after opening it!*

Alternative instruction: `InFile = open(filename, 'r', encoding='utf-8')`

Write File

Example of WRITING to a file

```
filename = input
("What is the name of the file to
open? ")

OutFile = open(filename, 'w')


for n in range(10):
    myFile.write('Number ‘ +
str(n))

OutFile.close()
```

open() *function, using the* ***‘w’*** *option means that we want to WRITE to this file. Note that* ***filename*** *is a string.*

This is the data that we’re creating to put into the file. Note that the use of the ***write()*** *function here means that the output goes to* ***“file output”*** *(not “standard output”)*
NOTE: ENTRIES HAVE TO BE STRING DATA TYPES!!!

Always ***close()*** *the file after opening it!*

To Reset Reading a File

- To go back to the start of a file that’s being read, you can use **open( )** again

- There are other (more sophisticated) ways to jump back and forth in reading/writing, but we’ll leave that for another class…

Demonstration

- **Given**: An input file with information on rainfall (in inches) for various geographical locations. Looks like this:

```
Akron 25.81
Albia 37.65
```

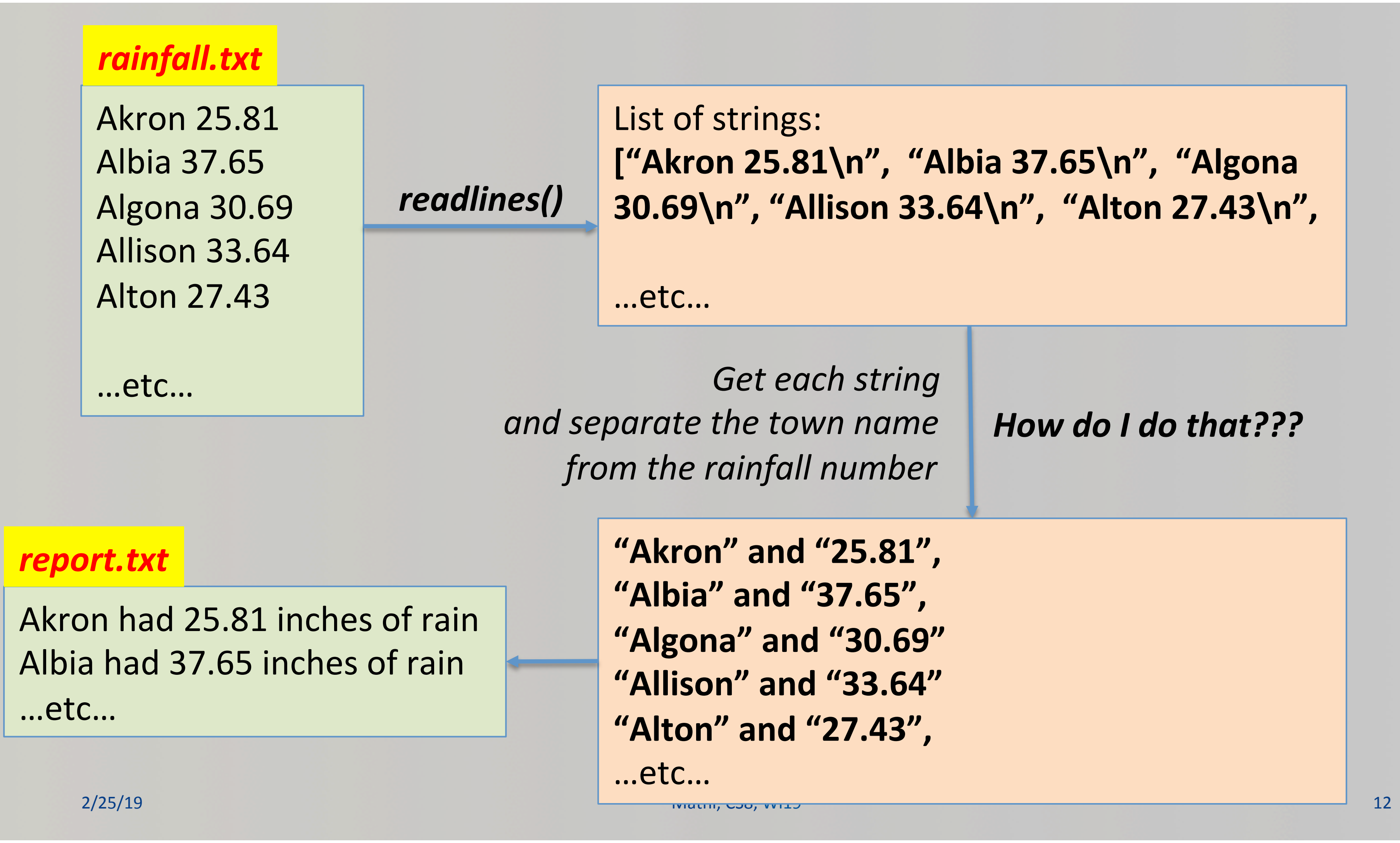
…etc…

- **You have to**: Create an output file that reads each line and outputs:

```
Akron had 25.81 inches of rain.
Albia had 37.65 inches of rain.
```

…etc…

See **rainfall.py** and **rainfall_advanced.py**



rainfall.txt

Akron 25.81
Albia 37.65
Algona 30.69
Allison 33.64
Alton 27.43

…etc…

readlines()

List of strings:
[“Akron 25.81\n”, “Albia 37.65\n”, “Algona 30.69\n”, “Allison 33.64\n”, “Alton 27.43\n”,

…etc…

Get each string
and separate the town name
from the rainfall number

How do I do that???

“Akron” and “25.81”,
“Albia” and “37.65”,
“Algona” and “30.69”
“Allison” and “33.64”
“Alton” and “27.43”,
…etc…

report.txt

Akron had 25.81 inches of rain
Albia had 37.65 inches of rain
…etc…

```
# Rainfall Example
# (c) 2017 by Ziad Matni for CS8

inputFile = open("rainfall.txt","r")
outputFile = open("report.txt", "w")

outputFile.write("Here's the rainfall report from around the nation!\n")
outputFile.write("---------------------------------------------------\n")

allLines = inputFile.readlines()

for line in allLines:
    values = line.split()
    outputFile.write(values[0]+" had "+values[1]+" inches of rain.\n")

inputFile.close()
outputFile.close()
```

```
# Rainfall Example
# WITH accumulated sum and average calculations
# (c) 2017 by Ziad Matni for CS8

inputFile = open("rainfall.txt","r")
outputFile = open("report.txt", "w")

outputFile.write("Here's the rainfall report from around the nation!\n")
outputFile.write("--------------------------------------------------\n")

allLines = inputFile.readlines()
count = 0
sum = 0

for line in allLines:
     values = line.split()
     outputFile.write(values[0]+" had "+values[1]+" inches of rain.\n")
     count += 1
     sum += float(values[1])

average = sum/count
inputFile.close()
outputFile.close()
```

NOT ON MIDTERM #2
(but still important)

Random Numbers

- "Pseudo-random" values can be generated using special functions in most programming languages
- In Python use functions of the **random module**
  - You have to ***import random*** first
- Simplest way to make a random number: **random.random()**
  - Returns a floating point value between 0.0 and 1.0
- Also: **randrange(n)**, **randint(low, high)** and many others
- Try typing **help(random)** in IDLE to learn more…
  - And play around with it

DEMO!

Question 1

Q: What is a Python statement that generates a number between 0 and 100 (**including** floating point values like 55.5)

Assume I issue a statement at first, like this:

```
from random import *
```

A. `random() + 100`

B. [✓ `random()*100`]

C. `random()/100`

D. `random(100)`

Question 2

Q: What is a Python statement that generates a INTEGER between 50 and 100. Assume you have the correct import statements…

A. `random() * 50`

B. `50 + int(random() * 50)`

C. `randrange(50, 100)`

D. `Both B and C` ←

E. `All of A, B, C`

F. `None of the above`

A Note for Lab 6

```
def rollDice():
'''
returns sum of rolling two six sided die'''

def rollDistribution(n):
'''
rolls a pair of die n times, returns the tally'''

def printDistribution(diceTally):
'''
prints the diceTally as a histogram'''
```

Midterm Exam #2

- Open Questions

YOUR TO-DOs

- □ **MIDTERM EXAM #2!**

- □ **HW7** (due on **Monday, 3/4**)
- □ **Lab6** (go to lab tomorrow)
- □ Keep working on your Project Assignment!

</LECTURE>