Computer Science 125
24 September 2020

1. How do we read a text file in Python?

```
F = open("numbers.txt", "r")
for line in F:
    print(line)
F.close()
```

Save this code as a Python file in the same folder that contains numbers.txt.

2. How can Python determine the number of lines in a text file?

```
F = open("numbers.txt", "r")
lines = F.readlines()      # returns a list of strings, each a line from F
print(len(lines))
F.close()
```

3. Use Python to write a file containing random numbers.

```
import random
F = open("newfile", "w")
for i in range(20):
    num = random.random()
    F.write(str(num))
F.close
```

```
import random
F=open("newfile.txt","w")
for i in range(20):
    num = random.random()
    F.write(str(num)+" ")
    if i % 5 == 4:
        F.write("\n")
F.close()
```

4. **Use Python to read a file of numerical data. Print the sum of the numbers on each line of the file. Then write the sums to a new file.**

```
F = open("numbers.txt", "r")
W = open("totals.txt", "w")
for line in F:
    items = line.split()    # returns a list of strings
    total = 0
    for i in items:
        total = total + int(i)
    print(total)
    W.write(str(total) + "\n")
F.close
W.close
```

PRACTICE WITH FILES

Working with a partner/group, use the following steps to solve each of the following problems.

(a) Plan your code on the white board (either on the classroom wall or on Zoom). Write out your entire program. Think about what errors might occur and how to fix them.

(b) Plan multiple test cases. What input will you send to your function? For each input, what value should be returned?

(c) *Only after you have completed steps (a) and (b) should you type your code in Python.*

(d) After you have typed your code, run your test cases. Does your code work? If not, how can you fix it?

1. Write a function that reads a text file (such as `milton.txt` on the course web site), converts each word to upper case, and saves the result to a new file. The new file should be exactly like the old file, including the same line breaks, but with all words in upper case. *Hint*: remember `str.upper()`

```
def toUppercase():
    F = open("milton.txt", "r")
    W = open("miltonUpper.txt", "w")

    for line in F:
        W.write(line.upper() + "\n")

    F.close()
    W.close()
```

2. Write a function that identifies the longest word(s) in a file. Your function should return the length of the longest word, along with all words of that length that occurred in the file. Make sure your function properly opens and closes the file.

```
def longestWords(filename):
    F = open(filename, "r")
    length = 0
    longwords = []

    for line in F:
        words = line.split()
        for w in words:
            if len(w) > length:
                length  = len(w)
                longwords = [w]
            elif len(w) == length and w not in longwords:
                longwords.append(w)

    F.close()

    return (length, longwords)
```

3. Write a function `joinFiles(inFile1, inFile2, outFile)` that accepts three file names. You function should read the contents of `inFile1` and `inFile2`, then write the contents of both to `outFile`. Make sure your function properly opens and closes each file.

   Extension: Modify your function to be `joinFiles(inFileList, outFile)`, which accepts a list of input file names, reads all of them, and writes the contents to `outFile`.

```
def joinFiles(inFile1, inFile2, outFile):
    F1 = open(inFile1, "r")
    F2 = open(inFile2, "r")
    W = open(outFile, "w")

    for line in F1:
        W.write(line)

    for line in F2:
        W.write(line)

    F1.close()
    F2.close()
    W.close()
```

4. **Bonus:** Find a dictionary file online that contains a huge list of words in the English language. Then write a program that reads the dictionary file and outputs words that contain all five vowels.