COMP 141

Using Loops to Read Files

Rhodes College

1

1

Announcements

- Program 5 due tomorrow
- Remote tutoring is available
  - Sunday-Thursday 5-11pm (CT)
- Join Slack
  - Course announcements, information, Zoom links, etc, will be put in Slack instead of email starting next week
- As always, check the Course Website for updates

2

Quiz 6

```
def main():
    for x in range(0, 6, 3):
        for y in range(x+2, 12, 2):
            print(y, end=' ')
        print()
main()
```

Solution:

```
2 4 6 8 10
5 7 9 11
```

3

Practice From Monday

- Write a program that writes a series of random numbers to a file. Each random number should be in the range of 1 through 100. Write at least 5 random numbers to the file – 1 number/line.
- Call your output file **`randomNums.txt`**

4

4

Practice from Monday Solution

generateRandomNumsFile.py in Box folder

```
import random

def main():
    #Open the file in write mode
    file = open("randomNums.txt", 'w')

    #You can change this for loop range to generate
    #as many random numbers as you'd like
    #This loop generates 100 numbers and writes each one
    #out to a file, 1 number per line
    for i in range(100):
        num = random.randint(1, 100)
        file.write(str(num) + "\n")
    file.close()

    #This is here solely so we can tell when the code is done running easily
    print("Done")

main()
```

5

Step 1: Open the file

- Uses the open() function.
- Always done the same way no matter how the file is organized.

```
file = open("filename.txt", "r")
```

open() returns a "file object," which is a data type like int, float, or string.

Replace this string with the real name of your file (don't forget the quotes!)

The "r" means open the file for reading.

6

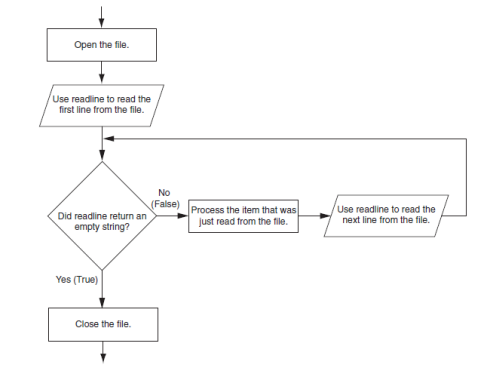
Using Loops to Read Files

- **Files typically used to hold large amounts of data**
  - Loop typically involved in reading from and writing to a file
- **Often the number of items stored in file is unknown**
  - The `readline` method uses an empty string as a sentinel when end of file is reached
    - Can write a while loop with the condition `while line != ''`

7

7

Figure 7-17 General logic for detecting the end of a file



8

8

Program 7-9 (read_sales.py)

```
# This program reads all of the values in
# the sales.txt file.

def main():
    # Open the sales.txt file for reading.
    sales_file = open('sales.txt', 'r')

    # Read the first line from the file, but
    # don't convert to a number yet. We still
    # need to test for an empty string.
    line = sales_file.readline()

    # As long as an empty string is not returned
    # from readline, continue processing.
    while line != '':
        # Convert line to a float.
        amount = float(line)

        # Format and display the amount.
        print(format(amount, '.2f'))

        # Read the next line.
        line = sales_file.readline()

    # Close the file.
    sales_file.close()

# Call the main function.
main()
```

sales.txt - Notepad

File Edit Format View Help

```
1000
2000
3000
4000
5000
```

Program Output

```
1000.00
2000.00
3000.00
4000.00
5000.00
```

9

9

Using Python's `for` Loop to Read Lines

- **Python allows programmer to write a `for` loop that automatically reads lines in a file and stops when end of file is reached**
  - Format: `for` *`line`* `in` *`file_object`*`:`
    `statements`
  - The loop iterates once over each line in the file

10

10

Program 7-10 (read_sales2.py)

```
# This program uses the for loop to read
# all of the values in the sales.txt file.

def main():
    # Open the sales.txt file for reading.
    sales_file = open('sales.txt', 'r')

    # Read all the lines from the file.
    for line in sales_file:
        # Convert line to a float.
        amount = float(line)
        # Format and display the amount.
        print(format(amount, '.2f'))

    # Close the file.
    sales_file.close()

# Call the main function.
main()
```

sales.txt - Notepad

File Edit Format View Help

```
1000
2000
3000
4000
5000
```

Program Output

```
1000.00
2000.00
3000.00
4000.00
5000.00
```

11

11

Step 2: Loop over the file

```
file = open("filename.txt", "r")
while [there are more lines in the
       file that we haven't read]:
  line = [read the next line
          from the file]
```

You only have access to one line of the file at a time.

12

12

Step 2: Loop over the file

```
file = open("filename.txt", "r")
for line in file:
```

line can be any string variable you want. This variable will store each line of the file as it is read.

13

Step 3: Process each line

- Do whatever you need to do with the string variable (usually called line).

```
file = open("filename.txt", "r")
for line in file:
  print(line)
```

14

Step 3: Process each line

- Usually a good idea to "strip" the newline character from the line before processing:

```
file = open("filename.txt", "r")
for line in file:
  line = line.rstrip()
  print(line)
```

15

Step 4: Close the file

- After you are done reading from the file, you should close the file:

```
file = open("filename.txt", "r")
for line in file:
  line = line.rstrip()
  print(line)
file.close()
```

16

Complete file-reading loop

- Use this as a template for file reading:

```
file = open("filename.txt", "r")
for line in file:
  line = line.rstrip()
  [process a line]
file.close()
```

17

Good Practice Tip

- When you are first reading from a file, you should **print** out each line as you get it so that you can see what is going on.
- When trying to debug your code, it's a good idea to add print statements so that you can understand what your program is doing at various steps.

18

18

Class Practice

Open your **randomNums.txt** file that you created last class and read in each number.

1. Write a program that outputs the sum and average of the numbers in your file**.**
2. Write a program that calculates the consecutive differences between numbers in the file
   (**Hint:** use the sliding window technique)
3. **Challenge**: Write a program to print out the smallest and largest values in the file.

- If you need a randomNums.txt file, I put mine in my Box.com code directory.
- Remember that the randomNums.txt file and your Python file need to reside in the same folder!

19

19

- Problem that re-occurs often in CS:
- Finding the largest item in a set of things where you can only look at each thing once.

20

- Pseudocode for finding the largest number in a collection of numbers:

- **`largest`** = [smallest possible number that you could ever see]
- look at each number once:
  if the current number > **`largest`**, then
  `largest` = current number

- after this loop, **`largest`** will have the largest number in it!

21