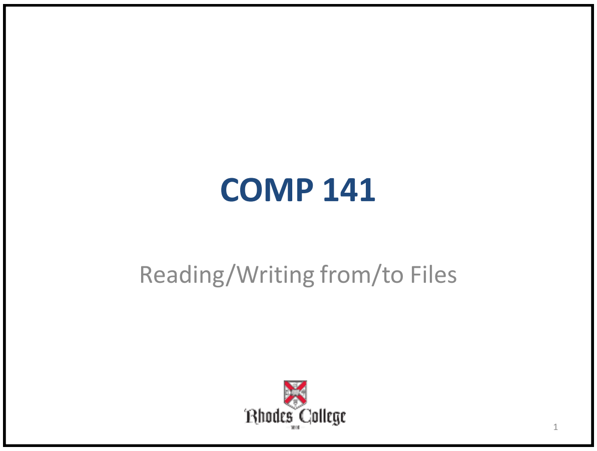
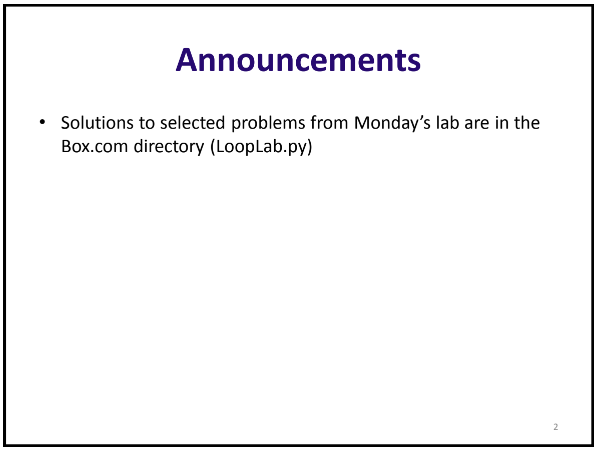


COMP 141

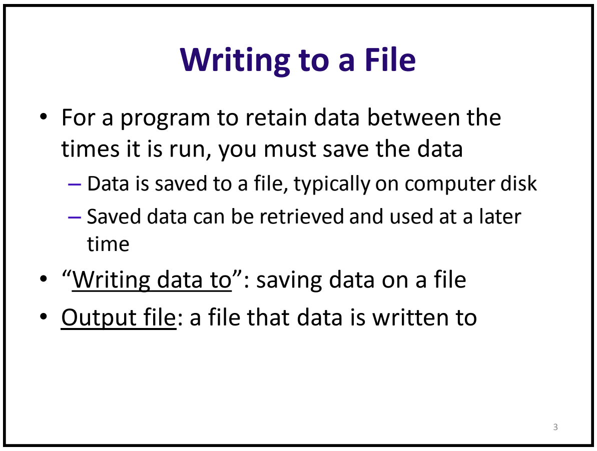
Reading/Writing from/to Files

Rhodes College



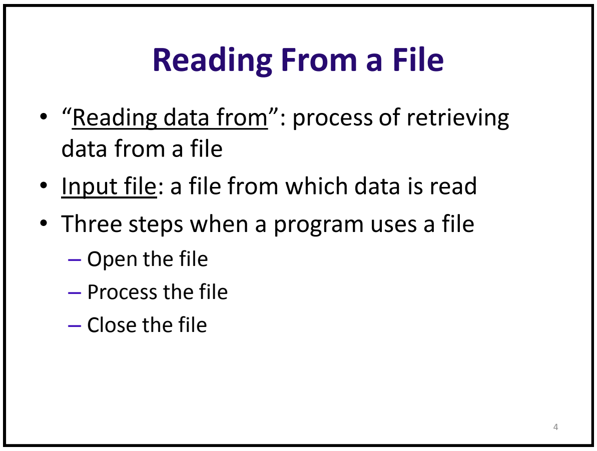
Announcements

- Solutions to selected problems from Monday's lab are in the Box.com directory (LoopLab.py)



Writing to a File

- For a program to retain data between the times it is run, you must save the data
  - Data is saved to a file, typically on computer disk
  - Saved data can be retrieved and used at a later time
- "Writing data to": saving data on a file
- Output file: a file that data is written to



Reading From a File

- "Reading data from": process of retrieving data from a file
- Input file: a file from which data is read
- Three steps when a program uses a file
  - Open the file
  - Process the file
  - Close the file

Types of Files and File Access Methods

- In general, two types of files
  - Text file: contains data that has been encoded as text
  - Binary file: contains data that has not been converted to text
- Two ways to access data stored in file
  - Sequential access: file read sequentially from beginning to end, can't skip ahead
  - Direct access: can jump directly to any piece of data in the file

Filenames and File Objects

- Filename extensions: short sequences of characters that appear at the end of a filename preceded by a period
  - Extension indicates type of data stored in the file
- File object: object associated with a specific file
  - Provides a way for a program to work with the file: file object referenced by a variable

Opening a File

- `open` function: used to open a file
  - Creates a file object and associates it with a file on the disk
  - General format:

    ```
    file_object = open(filename, mode)
    ```
- Mode: string specifying how the file will be opened
  - Example: reading only ('r'), writing ('w'), and appending ('a')

Specifying the Location of a File

- If `open` function receives a filename that does not contain a path, assumes that file is in same directory as program
- If program is running and file is created, it is created in the same directory as the program
  - Can specify alternative path and file name in the `open` function argument
    - Prefix the path string literal with the letter `r`

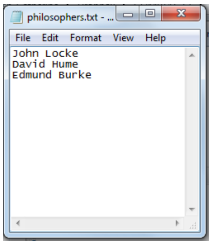
Writing Data to a File

- Method: a function that belongs to an object
  - Performs operations using that object
- File object's `write` method used to write data to the file
  - Format: *`file_variable`*`.write(`*`string`*`)`
- File should be closed using file object `close` method
  - Format: *`file_variable`*`.close()`

Writing Data to a File

Program 7-1 (file_write.py)

```
# This program writes three lines of data
# to a file.
def main():
    # Open a file named philosophers.txt.
    outfile = open('philosophers.txt', 'w')

    # Write the names of three philosphers
    # to the file.
    outfile.write('John Locke\n')
    outfile.write('David Hume\n')
    outfile.write('Edmund Burke\n')

    # Close the file.
    outfile.close()

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



```
John Locke
David Hume
Edmund Burke
```

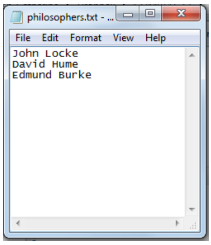
Reading Data From a File

- `read` method: file object method that reads entire file contents into memory
  - Only works if file has been opened for reading
  - Contents returned as a string
- `readline` method: file object method that reads a line from the file
  - Line returned as a string, including '`\n`'
- Read position: marks the location of the next item to be read from a file

Reading Data From a File

Program 7-2 (file_read.py)

```
# This program reads and displays the contents
# of the philosophers.txt file.
def main():
    # Open a file named philosophers.txt.
    infile = open('philosophers.txt', 'r')

    # Read the file's contents.
    file_contents = infile.read()

    # Close the file.
    infile.close()

    # Print the data that was read into
    # memory.
    print(file_contents)

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



```
John Locke
David Hume
Edmund Burke
```

Program Output

```
John Locke
David Hume
Edmund Burke
```

Reading Data From a File

```
# This program reads the contents of the
# philosophers.txt file one line at a time.
def main():
    # Open a file named philosophers.txt.
    infile = open('philosophers.txt', 'r')

    # Read three lines from the file
    line1 = infile.readline()
    line2 = infile.readline()
    line3 = infile.readline()

    # Close the file.
    infile.close()

    # Print the data that was read into
    # memory.
    print(line1)
    print(line2)
    print(line3)

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

Program Output

```
John Locke

David Hume

Edmund Burke
```

Concatenating a Newline to and Stripping it From a String

- In most cases, data items written to a file are values referenced by variables
  - Usually necessary to concatenate a '`\n`' to data before writing it
    - Carried out using the + operator in the argument of the `write` method
- In many cases need to remove '\n' from string after it is read from a file
  - `rstrip` method: string method that strips specific characters from end of the string

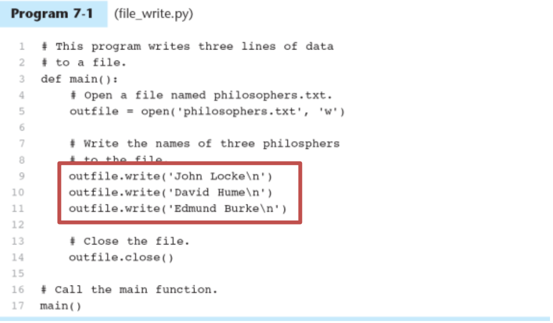
Stripping Newline from a String

```
# This program reads the contents of the
# philosophers.txt file one line at a time.
def main():
    # Open a file named philosophers.txt.
    infile = open('philosophers.txt', 'r')

    # Read three lines from the file
    line1 = infile.readline()
    line2 = infile.readline()
    line3 = infile.readline()

    # Strip the \n from each string.
    line1 = line1.rstrip('\n')
    line2 = line2.rstrip('\n')
    line3 = line3.rstrip('\n')

    # Close the file.
    infile.close()

    # Print the data that was read into
    # memory.
    print(line1)
    print(line2)
    print(line3)

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

Program Output

```
John Locke
David Hume
Edmund Burke
```

Writing Data to a File



Program 7-1 (file_write.py)

```
# This program writes three lines of data
# to a file.
def main():
    # Open a file named philosophers.txt.
    outfile = open('philosophers.txt', 'w')

    # Write the names of three philosphers
    # to the file.
    outfile.write('John Locke\n')
    outfile.write('David Hume\n')
    outfile.write('Edmund Burke\n')

    # Close the file.
    outfile.close()

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

Appending Data to an Existing File

- When open file with ‘w’ mode, if the file already exists it is overwritten
- To append data to a file use the ‘a’ mode
  - If file exists, it is not erased, and if it does not exist it is created
  - Data is written to the file at the end of the current contents

Writing and Reading Numeric Data

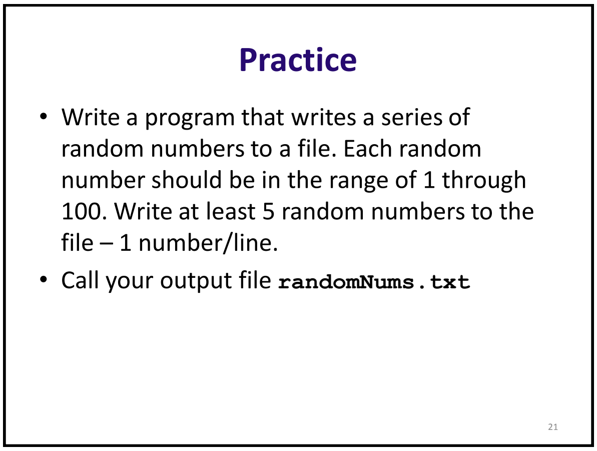
- Numbers must be converted to strings before they are written to a file
- `str` function: converts value to string
- Number are read from a text file as strings
  - Must be converted to numeric type in order to perform mathematical operations
  - Use `int` and `float` functions to convert string to numeric value

Reading Numbers from a File

```
# This program demonstrates how numbers that are
# read from a file must be converted from strings
# before they are used in a math operation.

def main():
    # Open a file for reading.
    infile = open('numbers.txt', 'r')

    # Read three numbers from the file.
    num1 = int(infile.readline())
    num2 = int(infile.readline())
    num3 = int(infile.readline())

    # Close the file.
    infile.close()

    # Add the three numbers.
    total = num1 + num2 + num3

    # Display the numbers and their total.
    print('The numbers are:', num1, num2, num3)
    print('Their total is:', total)

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

Writing Numbers to a File

```
# This program demonstrates how numbers
# must be converted to strings before they
# are written to a text file.

def main():
    # Open a file for writing.
    outfile = open('numbers.txt', 'w')

    # Get three numbers from the user.
    num1 = int(input('Enter a number: '))
    num2 = int(input('Enter another number: '))
    num3 = int(input('Enter another number: '))

    # Write the numbers to the file.
    outfile.write(str(num1) + '\n')
    outfile.write(str(num2) + '\n')
    outfile.write(str(num3) + '\n')

    # Close the file.
    outfile.close()
    print('Data written to numbers.txt')

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



Practice

- 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`**