File I/O

CS303E: Elements of Computers
and Programming
July 23, 2012

Announcements

Exam returned on tomorrow

BIG AI

Open Source Software

- *Open source* refers to a class of software that distributes the code (and binaries) for free
  - Anyone can modify
- The open source community refers to the programmers that develop this software
  - Anyone can join (you too!)

Open Source: Sample Projects

- Openoffice
- Linux
- Mozilla/Firefox
- Ushahidi

Programming for the Open Source Community

Some open source projects are open to student involvement
- SoftHum (Software for Humanity) has a long list of potential student projects
  - Other websites also relate to this HumIT, HFOSS
- Also has information about how to evaluate open source projects before committing

File I/O

- Files have different types
  - Text
  - Binary
- I/O: Input/Output
- Steps:
  - Open file
  - Read from and/or write to the file
  - Close the file

Opening a File

- When you open a file, you open it in a particular *mode*:
  - "r" to read
  - "w" to write
  - "a" to append
- *write* mode deletes any existing file with that name and then writes
- *append* mode begins writing at end of any existing file with that name

Opening a File: Syntax

```
myFileVar = open("myFileName",
  "theMode")
```

To read from file "myData.txt":

```
inFile = open("myData.txt","r")
```

Closing a File

A file must be closed after you are finished reading or writing
– Otherwise, changes may not be saved to disk

Syntax:
```
myFileVar.close()
```

Writing To a File

- Use `write()` to add text to a file
  – Accepts a string as its parameter
  – Similar to output to the screen
  – Same string formatting
- BUT always need to tell the computer to change lines
  – How?

Writing to a File: Example

```
myFileVar = open("data.txt","w")
myFileVar.write("First line\n")
myFileVar.write("Second line")
i = 2
myFileVar.write("Also line %d\n" % i)
myFileVar.close()
```

Output:
```
First line
Second lineAlso line 2
```

Exercise

Write a program that opens a file for writing and writes the numbers 1 to 10, one per line, in a new file. The program should also close the file. Remember: the `write()` command takes a string as its parameter.

iClicker Question

Which `write()` command is valid? Assume `inFile` is a valid file.

A. `infile.write(42)`
B. `infile.write("%d"% 42)`
C. `infile.write(The meaning of life is 42)`

Reading From a File

- `read()`
  - Returns a string containing the entire file
- `readlines()`
  - Returns a *list* where each element is one of the remaining lines in the file
- `readline()`
  - Returns a string containing the next line of the file (including the ending newline)
  - Works on large files

Reading from a File: Single Line to Whole File

To read from the beginning of the file to the end:
  - Iterate over it!

```
for line in myFileVar:
   <code>
```

Reading From a File: Gotcha!

Lines will end with a newline character and may have other "hidden" white space

For example, the lines:
```
  Mary had a little lamb
  Whose fleece was white as snow
```
May actually be:
```
\tMary had a little lamb\n
\tWhose fleece was white as snow\n
```

Exercise

Write a program that opens a file for reading and then prints each line in the file to the screen twice in uppercase. The program should also close the file.

iClicker Question

What is at the end of every line read in from a file?

A. "\t"
B. "\\"
C. "\n"

Iterating Over a File: Other Ways

Other ways to iterate over a file:

```
for line in inFile.readlines():
  <code>

line=infile.readline()
while(len(line)!=0)     #at EOF returns ""
  <code>
  line=infile.readline() #update loop
                         #  variable!
```

Files are Objects

- *Objects* are a grouping of data and operations that can be performed on that data
  - The operations are essentially functions, but in this context they are called *methods*
- To call a method on an object:
  - `objectName.methodName()`

Examples

```
fileVar.read()
fileVar.open()
stringVar.upper()
stringVar.find(sub)
```

iClicker Question

When you finish using a file, it is important to close it.

A. True
B. False

Files as Parameters

- Recall that the file object maintains state for you, so that each time you call `readline()` you receive the next line
- This state is maintained even as the file is passed into functions as a parameter
- AND changes to that state are reflected in the calling function after the call

Exercise

Using the IMDB list of top movies, "TopMoviesIMDB.txt", find the number of titles that have "cat" or "dog" in them. Print each title to the file "petMovies.txt".

iClicker Question

When you finish using a file, it is important to close it.

A. True
B. False