Writing Files

10-18-2010

Opening Discussion

- Solutions to the interclass problem.
- Any questions about the assignment?
- Have you looked at the reading?
- Minute essay comments:
  - Dimensionality of arrays.
  - Why can't we just talk to computers?
  - Delimiters?

java.util.Scanner

- Java Scanner class sometimes easier for input.
  - hasNext(), next()
  - hasNextInt(), nextInt()
  - hasNextDouble(), nextDouble() ...
- Doesn't produce a Scala collection.
- Needs java.io.File: new Scanner(new File(fileName))
- Always make a new object from the Java libraries using new. Scala typically allows you to leave that off.

Closing Files

- Make sure you always close files when you are done using them.
- Source, Scanner, and pretty much anything else that pulls from a file will have a close() method.

java.io.PrintWriter

- To write to files use java.io.PrintWriter.
- Create with new PrintWriter(fileName)
- Has print and println methods just like what you have been using to print to screen.

Flush

- To make certain contents have been written to the file use flush().
- Doing close() will also flush and you should definitely remember to close all files you are writing when done with them.

Command Line Arguments

- In a script, the command line arguments are put in args:Array[String].
- You can do anything with them that you would do with a normal array.

Examples

- Print some random numbers to a file.
- A file copy with word replace.

Minute Essay

- Have any questions?
- Interclass problem:
  - Write a script that will read in a file of numbers. The file can have multiple numbers on each line. It should write out to a different file. Each line in the new file should have the min, max, and mean of the numbers that were on that line in the input.