

# Writing Files

3-2-2011

# Opening Discussion

- Minute essay comments:
  - Why are Mondays so awful?
  - Why use an Iterator if it gets consumed after one use?
  - Starting to see the power of Scala.
- Solutions to the interclass problem.

# 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?
- What can I do that would help you with the material? Are you reading and trying to do exercises and other things from the book?