

### Administrivia

- Reminder — Homework 7 code due today. Okay to turn in plots next time in class. Notice that you were to implement your heap *with an array-like data structure*, not as an explicit tree.
- Due dates for Homework 8 changed (design next Tuesday, code same day as final). Writeup coming soon.
- For the final, two options — let's discuss, and then vote by e-mail:
  - Straight written exam.
  - Written exam plus game presentations.

Slide 1

### I/O in Java, Recap

- I/O libraries in Java are very powerful, but some significant differences from C/C++ — more like "connecting a series of pipes".
- Many operations require at least two classes — one to connect to source/destination and one to provide functionality needed in program. Very easy to add "filter" (e.g., for encryption) as intermediate stage.
- "Serialization" — `Object*Stream` classes — is very powerful, but a bit risky if the implementation of the class might change. (To see contents of a binary file in Linux, `od` command.)
- Let's do another example — C-like input (read an integer, a double, a string, etc.).

Slide 3

### Minute Essay From Last Lecture

- Try writing code to count the lines of a file containing character data. (No need to make a complete class or method.)

It would look something like this:

```
BufferedReader rdr =
    new BufferedReader(new FileReader("whatever"));
String line;
int lines = 0;
while ((line = rdr.readLine()) != null)
    ++lines;
```

Slide 2

### Minute Essay

- None — quiz 6.

Slide 4