

# I/O Streams

9-21-2011

# Opening Discussion

- Minute essay comments
  - Does par automatically adjust for number of cores?
  - Other shortcuts for parallel?

# Streams

- Data goes passed like water in a stream.
- This is different from random access data where you can look anywhere you want at any time.
- Streams are more general because they don't allow you complete access.

# java.io Package

- InputStream and OutputStream form bases of trees that work with bytes.
- Reader and Writer form bases of trees that work with chars.
- File streams access files.
- File class is very handy.

# Wrapping Streams

- You can wrap one stream in another to provide additional functionality.
- Buffered types improve performance.
- `DataInputStream` and `DataOutputStream` allow binary data.

# Serialization

- `ObjectInputStream` and `ObjectOutputStream` allow serialization.
- This writes an object to a stream or reads an object from a stream.
- Scala also has a pickling library we might discuss later.
- You have to add annotations.
  - `@serializable` – on the class
  - `@SerialVersionUID(123)` – on the class
  - `@transient` – on members

# Code

- Let's spend the rest of our time putting some things into the code.
- Ideally we'll get it so we can do a bit of drawing and save the drawings.

# Minute Essay

- What is the primary advantage of the Java stream model?
- Look at the “in-class code” and make sure you understand it.
- Quiz #2 is next class.