READING FROM FILES.

- In Java we utilize **streams** to read from and write to files.

- A stream is simply a **sequence of data**.

- Java has a delivered set of input streams which all inherit from a base `InputStream` class.
  - [http://docs.oracle.com/javase/7/docs/api/java/io/InputStream.html](http://docs.oracle.com/javase/7/docs/api/java/io/InputStream.html)

- Using inheritance and polymorphism, we can treat these streams interchangeably.
READING FROM FILES.

- We have already seen an example of reading data from a stream using a Scanner object.
- Reading data from the user console is one type of stream.
- To read from a file is the same, but utilize a different stream.

WRITING TO FILES.

- In Java we utilize streams to write to files.
- Java has a delivered OutputStream classes.
  - [http://docs.oracle.com/javase/7/docs/api/java/io/OutputStream.html](http://docs.oracle.com/javase/7/docs/api/java/io/OutputStream.html)
- Using inheritance and polymorphism, we can treat these streams interchangeably, although some have specialized functionality.
JAVA SERIALIZABLE INTERFACE

• From the Java APIs:

Serializability of a class is enabled by the class implementing the java.io.Serializable interface.
Classes that do not implement this interface will not have any of their state serialized or deserialized.
All subtypes of a serializable class are themselves serializable. The serialization interface has no methods or fields and serves only to identify the semantics of being serializable.