File I/O Questions

- What are the Java classes for dealing with binary files containing Java types and objects?
  - ObjectInputStream
  - ObjectOutputStream
- What are the basic Java classes for dealing with binary files?
  - FileInputStream
  - FileOutputStream
- What is a "binary file"?
  - Trick answer: All files contain binary data.
  - Useful answer: A file that contains other than just ASCII characters (or "unusual" ASCII control characters).
- What are the Java classes for dealing with text files?
  - Scanner(File)
  - PrintWriter(File)
  - But, System.out is a PrintStream(File)
- How are alternative, larger (international) character sets represented?
  - Unicode and Unicode Transformation [Text?] Format (UTF)
  - UTF-8: 8-bit version, essentially ASCII
  - UTF-16: two bytes (or more) per character
- What are characters and strings and how are they written to a file/stream? [US-centric answer]
  - In Java, characters are represented as two bytes (16 bits). The actual character set is language dependent. In the US, we use ASCII, which only takes 7 bits, leaving one byte always zero in Unicode.
  - Strings are sequences of characters.
  - In a text file, characters are stored in successive bytes in the file. For plain text, ASCII files, the leading (zero) byte is omitted--leaving one byte per character.
- How have we seen these abstractions generalized still further in Java threads?
  - With producer/consumer threads
- How is reading from a console like reading from a file?
  - Reading gets the next character/word/line from the console or "current location" in the file.
- How is printing to the console like writing to a file?
  - Output is appended to the "current location" in the file or on the console.
- How are other I/O devices similar ? (printer, network connection)
  - They all can be represented by a "stream" of characters (or bytes or objects) that can be read from or appended to.
- How are these abstractions generalized in, say, UNIX?
  - Processes read from stdin, write to stdout.
  - Fundamental abstraction is a stream of characters (or bytes or data objects).
  - Streams may be "linked" to files, keyboard, console, network connection, or other processes.
  - Pipes link the stdout of one process to the stdin of another
  - Angle brackets allow stdin to come from a file (< inputFile) or stdout to go to a file (> outputFile).