Byte-based I/O

22 Feb 2008
CMPT166
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Review last time

- Exceptions in I/O
- Serializable objects
- Object-based I/O: ObjectInputStream
- Random-access files

Random-access files

- Sequential files are hard to modify in-place
  - Must erase and rewrite entire file
- Random-access files:
    - file = new RandomAccessFile(“user.db”, “rw”);
- Can be used in place of FileInputStream / FileOutputStream, e.g., to do object-based I/O
- File position pointer:
    - file.seek(num_bytes);
  - Seek to position relative to start

Classes for byte-based I/O

- OutputStream: abstract class for byte-based I/O
  - FileOutputStream: subclass of OutputStream
  - ObjectOutputStream: wrapper for objects
  - PipedOutputStream: between threads
- FilterOutputStream: filter/aggregate data
  - PrintStream: text output to the stream
    - System.out, System.err
  - DataOutputStream: byte output
- Also Input versions of all these

Interfaces for byte-based I/O

- DataOutput: writing primitive types to a stream
  - Implemented by class DataOutputStream
  - Also implemented by class RandomAccessFile
  - .write(), .writeBoolean(), .writeChar(), .writeChars(), .writeFloat(), .writeInt(), etc.
- ObjectOutput: writing objects to a stream
  - Implemented by class ObjectOutputStream
  - .writeObject()
- Also Input versions of all these

Buffered streams

- A buffer is intermediate storage for reads/writes before they are committed
- Speed/efficiency: hard-disk has high latency, so accumulate multiple I/O and execute as a group
  - Writes might not happen right away!
    - What happens in a power outage?
- BufferedOutputStream
  - Subclass of FilterOutputStream
  - .flush(): returns only after I/O is completed