Input / Output

Java I/O is built upon Streams

> Two type I/O streams:

Byte Streams

Character Streams

> In general, the functionality of byte streams is paralleled by that of character streams

> At the lowest level, all I/O is byte-oriented

Byte Stream Classes

> Two abstract class hierarchies define byte streams

- InputStream
- OutputStream

> Sub classes include:

- BufferedInputStream
- BufferedOutputStream

- PrintStream
  print() and println()

- FileInputStream
- FileOutputStream

etc

see table 10-1

Character Stream Classes

> Two abstract class hierarchies define character streams

- Reader
- Writer

> Sub classes include:

- BufferedReader
- BufferedWriter

- PrintWriter
  print() and println()

- FileReader
- FileWriter

etc.

see table 10-2

Predefined Streams

> Contained in java.lang is a class called System

> In System are three predefined stream variables
  - in
  - out all are defined as public, final, static
  - err thus useable anywhere

> System.out defaults to the console (redirection possible)
> System.in defaults to the keyboard

> System.in is an object of type InputStream (a byte stream)
> System.out is an object of type OutputStream (a byte stream)

Methods defined by InputStream

> int available() Returns number of bytes available for reading

> void close()

> int read() Returns an integer representation of next available byte. Returns -1 for end of stream found

> int read(byte buffer) Reads buffer.length # bytes into buffer

etc

throws IOExeption

Methods defined by OutputStream

- void close() Closes output stream. IOException if more output
- int write(int b) Write one byte to output. Converts int to byte
- int write(byte buffer[ ]) Writes a complete array of bytes to output stream

etc

```
// Read an array of bytes from the keyboard.

import java.io.*;

class ReadBytes {
  public static void main(String args[])
    throws IOException {
      byte data[] = new byte[10];

      System.out.println("Enter some characters.");
      System.in.read(data);
      System.out.print("You entered: ");
      for(int i=0; i < data.length; i++)
        System.out.print((char) data[i]);
  }
}
```

```
import java.io.*;

class ShowFile {
  public static void main(String args[])
  {
    int i;
    FileInputStream fin;
    try {
      fin = new FileInputStream(XXXX);
    } catch(FileNotFoundException exc) {
      System.out.println("File Not Found");
      return;
    }
    try {
      // read bytes until EOF is encountered
      do {
        i = fin.read();
        if(i != -1) System.out.print((char) i);
      } while(i != -1);
    } catch(IOException exc) {
      System.out.println("Error reading file.");
    }
    try {
      fin.close();
    } catch(IOException exc) {
      System.out.println("Error closing file.");
    }
  }
}
```