Networking and I/O

powerful, flexible

→ steep learning curve

Frequently "wrapping" objects
Suppose we want to read from file "input.txt".

```java
FileReader fr = new FileReader("input.txt");
while ((int cc = fr.read()) != -1) {
    System.out.println(cc);
}
```
```c
int ln = 0;
int cc;

FILE *fr = NULL;
while ((cc = fgetc(fr)) != EOF) {
  if (cc == 'M') {
    printf("Found line \"n\" + b.\n");
  } //exit
}```
FileReader fr = new FileReader("input.txt");

LineNumberReader in =
    new LineNumberReader(fr);

LineNumberReader in =
    new LineNumberReader(new FileReader("input.txt"));
int c;

while ( (c = in.read()) != EOF )
{
    if ( c == 'M' )
    {
        System.out.println( in.getLineNum() );
        System.exit( 0 );
    }
}
LineNumberReader ln = new
LineNumberReader(
    new Reader(new NetworkSocketReader(
        new BancyDrumReader("COM1"))));
System.out.println("Bleh, blah");

in my code
PrintStream out;
created PrintStream object....
out.println("Here's some output");
PrintStream out;

if (user wants to print to screen) {
    out = System.out;
} else if (user wants file) {
    out = new FileOutputStream("output.txt");
} else

................
System.in — static object in class System

Char c = System.in.read();
BufferedReader in = new BufferedReader(System.in);
String input = in.readLine();