More fun with objects and pointers
and a bit more on exceptions, too

About buffering ... why?
Palindrome and Spelling Suggester buffered results ... what’s the point?

java.util.Scanner also buffers results ... why?

Schedule reminder
Assignment 8 due Monday, 29 Nov
Optional assignment due Friday, 3 Dec
Optional midterm re-do Wednesday, 1 Dec
(or learn about Eclipse from CIS 210 GTFs)

Final exam Monday, 6 December, 3:15pm
tell me now if you have a conflict

Buffering in java.util.Scanner

What if it didn’t provide this? What would your file reading loop look like?
Lists versus Arrays:
What are lists good for?
What are they bad at?
Consider the array of Muppets a few assignments ago ...

Suppose I want to ...
Add a Muppet at the beginning of the sequence
Which is faster, array or list?
Add a Muppet at the end of the sequence
Which is faster, array or list?
Add a Muppet in alphabetical order by name
Which is faster, array or list?
Remove a Muppet from the middle of the sequence

Insertion at front of list

First we have to find the end ... typically search the whole list
We say it is a linear time operation (proportional to list length),
or O(n) where n is length of list.
( “big-oh of n” )
Delete from list

Cheap (constant time) once we find the element to delete ... but finding it still takes linear time.

Are arrays better or worse?

Exceptions

How to handle them

Ex: FileNot Found Exception

How to create them

Ex: StackUnderflowException, TooManyMuppetsException

Rules of design with exceptions

Handling an Exception

Scanner console = new Scanner( System.in );
Scanner infile = null;
String fname = args[0];
while (infile == null) {
    try {
        infile = new Scanner ( new File ( fname ) );
    } catch (FileNotFoundException e) {
        System.out.println(e);
        System.out.println("Let's try that again, shall we?");
        System.out.println("File name> ");
        fname = console.next();
    }
}
System.out.println("Opened "+ fname + " successfully");

Try it ...

$ java Exceptions nosuchfile.txt
java.io.FileNotFoundException: nosuchfile.txt (No such file or directory)
Let's try that again, shall we?
File name> really?
java.io.FileNotFoundException: really? (No such file or directory)
Let's try that again, shall we?
File name> words.txt
Opened words.txt successfully
Exceptions are objects

class PitchException extends Exception {
    ...
    static void pitcher(String m) throws PitchException {
        if (m.equals("Nope")) {
            throw new PitchException();
        }
        System.out.println("Pitching: " + m);
    }
}

Why exceptions?

Experience with C:
    Programmers forget to check method results for failure.
    Programs crash in mysterious ways.

Remedy:
    You must catch or declare checked exceptions
    Java won’t let you forget!

Try it ...

    static void catcher() {
        try {
            pitcher("OK");
            pitcher("Nope");
            pitcher("Uh huh");
        } catch (PitchException e) {
            System.out.println("Caught exception: "+ e);
        }
    }

    static void pitcher(String m) throws PitchException {
        if (m.equals("Nope")) {
            throw new PitchException();
        }
        System.out.println("Pitching: " + m);
    }

Exception design rules

Use exceptions for **exceptional conditions**
    Simplify program logic by separating “what if” logic from normal cases

**Never** use exceptions for normal control flow
    Example: Not finding a muppet is not an exceptional condition
    The Java compiler is designed to make exceptions cheap **unless you throw them**