Announcements

• Wed night: program #4 due
• Sat night OWL #7 due
• Trapdoor news - 390 is magic number...
• Diffy problem, program #5 up soon
• Extra credit programming projects coming..
Arrays - part II

We are studying the arrays data structure

Cells welded together..can hold primitives or objects

How are they

• declared
• accessed
• traversed
• used for modeling
• generalized
Variables in algebra

\( x_0 \, x_1 \, y_3 \) and so forth

Java notation just a variant:
\( x[0], \, x[1], \, y[3] \)

Algebra: \( x_0 = 2 \times x_1 \)

---

Java: \( x[0] = 2 \times x[1]; \)
Defining array variables

```java
int[] nums = new int[6];  //array of 6 ints
nums[3] - the third one; at index 3
```

```java
Infant[] kids = new Infant[5];  //5 Infant
kids[0] - the zeroth one; at index 0
```

Indexing system like char/string
Let's initialize a list of numbers with five values:

```
nums = [0, 0, 0, 0, 0, 0]
```

Then, we set the value at index 3 to 8:

```python
nums[3] = 8
```

Now, the list looks like:

```
nums = [0, 0, 0, 8, 0, 0]
```
public class ArrayTest{
    public static void main(String[] args){
        int[] firstArray = new int[10];
        for(int j = 0; j < 10; j++){
            firstArray[j] = j*j;
        }
        System.out.println("here they come");
        for(int j = 0; j < firstArray.length; j++)
            System.out.println(firstArray[j]);
    }
}
Arrays - the mental picture..

```java
int[] firstArray = new int[10];

firstArray[6] = 17;
```

```
0 1 2 .......... 6 7 8 9
```

firstArray.length -> 10
These sorts of expressions are possible:

```plaintext
firstArray[4] = 9*firstArray[4];
firstArray[3] = 11;

int j = firstArray[3]/2;
firstArray[j] = 9*firstArray[j/2];
```
Shorthand

```java
int[] nums = {2, 4, 6, 8, 10};

makes an array of 5 ints:
System.out.println(nums[4]); -> prints 10
```

If myKid, yourKid, jillsKid, leahsKid, nedsKid already exist as Infant objects, this is ok:

```java
Infant[] someKids =
    {myKid, yourKid, jillsKid, leahsKid, nedsKid};
```
Arrays are objects

When you say “length” you are invoking a constant (public final value) associated with the array.

The size of an array is determined when “new” is invoked:

```java
int[] someArray = new int[66];

int[] nums; // this is ok - variable is named
```

Array indices always int, and always start at 0

Array indices end at cell # (length - 1): same as String indexing
An application

We’re going to write an application that rolls a pair of dice some number of times and reports the results as a profile of the rolls (e.g. how many 2, 3, 4, .. etc. came up).

The array as scoreboard
Results: (10,000 tosses)

toss of 2 303

toss of 3 543

toss of 4 807

toss of 5 1123

toss of 6 1432

toss of 7 1630

toss of 8 1389

toss of 9 1129

toss of 10 808

toss of 11 557

toss of 12 279
Key idea:
Indices of a “scoreboard” array actually stand for dice toss outcomes
import javax.swing.JOptionPane;

public class DiceExperiment {
    public static void main(String[] args) {
        String tossString = JOptionPane.showInputDialog("enter toss count");
        int tossCt = Integer.parseInt(tossString);
        Dice d = new Dice(tossCt);
        d.multiToss();
        d.showScoreboard();
    }
}
public class Dice{
    private int[] scoreboard = new int[13]; // complex!
    private int tossCt;

    public Dice(int tosses)
    { tossCt = tosses;
        initializeScoreboard();
    }

    public void initializeScoreboard(){ // why 0,1?
        for(int j = 0; j < 13; j++) scoreboard[j] = 0;
    }

    public int tossDie()
    { return (1+ (int)(6*Math.random()));
    }
}
public int throwDice()
{
    return(tossDie() + tossDie());
}

public void multiToss(){ // key method
    int score;
    for (int j = 0; j < tossCount; j++){
        score = throwDice();
        scoreboard[score]++;
    }
}

// Note: cells 0, 1, never get used
public int[] getScoreboard(){return scoreboard; }

public void showScoreboard(){
    for(int j = 2; j < 13; j++)
        System.out.println("toss of "+ j +" "+
                          scoreboard[j]);
}
} // ends class

Note: we ignore cells 0 and 1
Suppose $\text{throwDice()}$ returns 6 \( \rightarrow \) then what?
Arrays of objects

Infant[] kids = new Infant[10]; // array of 10 infants

Infant littleMikey = new Infant(“Mike”, 3);
kids[4] = littleMikey;
//Places littleMikey into cell with index 4 of kids array, via reference.

Kid at cell 2 has wrong name; should be Lilly
kids[2].setName(“Lilly”);

public void allOlder(Infant[] kiddo){
   for(int j = 0; j < kiddo.length; j++)
      kiddo[j].anotherMonth();
}
Typical array problems:

Find the name of the oldest kid in an array of Infants

Assume zeroth kid is the oldest - set aside her position (0), and her age

Walk down the array (may as well start with 1, but could start at 0 - this may have some advantages)
When you find someone older:
set aside her position (j), her age

When you're done, j holds the index of the oldest kid (who could be older???)

Get that kid, return her name!
An easier problem first..
public boolean anyBig(int[] nums){
    // are any nums > 2
    boolean aBig = false;
    for(int j = 0; j < nums.length; j++){
        if (nums[j] > 2){
            aBig = true;
            break;
        }
    }
    return aBig;
}

// note that one method fits ANY int array of // any size
public boolean anyBabies(Infant[] kiddo){
    // are any kids less than 2 months old?
    boolean aBaby = false;
    for(int j = 0; j < kiddo.length; j++){
        if (kiddo[j].getAge() < 2){
            aBaby = true;
            break;
        }
    }
    return aBaby;
}

// note that one method fits ANY Infant
// array of any size
public boolean majorityOld(Infant[] kiddo, int a) {
  // are strict majority in array older than a?
  int old = 0;
  for (int j = 0; j < kiddo.length; j++) {
    if (kiddo[j].getAge() > a) {
      old++;
    } else old--;
  }
  return (old > 0);
}
int theArray = {3,5,7,9,11};

myName.length() vs. theArray.length
int theArray = {3,5,7,9,11};

myName.length() vs. theArray.length

A String: length() method! An array, length a constant - no parens!
How can an array be a method parameter if you don't know how long it is?

When is each of these legal, where j is some int $\geq 0$, theArray is an array of ints?

```java
theArray[j] = theArray[Math.sqrt(j)];
theArray[j] = theArray[(int)Math.sqrt(j)];
theArray[j] = theArray[j+1];
theArray[j] = theArray[myName.length()];
```
How can an array be a method parameter if you don’t know how long it is?

The actual parameter is an object with a length field, and you can get at it via theArray.length

When is each of these legal, where j is some int >= 0

theArray[j] = theArray[Math.sqrt(j)]; never
theArray[j] = theArray[(int)Math.sqrt(j)]; in bounds
theArray[j] = theArray[j+1]; in bounds
theArray[j] = theArray[myName.length()]; in bounds
What's wrong with this picture?

```java
for(int j = 0; j <= theArray.length; j++)
    System.out.println(theArray[j]);
```
What’s wrong with this picture?

```java
for(int j = 0; j <= theArray.length; j++)
    System.out.println(theArray[j]);
```

`j` takes on value `theArray.length` - and there’s nothing there: you’ll get an `ArrayOutOfBounds Exception` (error)
An application:

Read in multiple lines of text - up to 50 - and then print the lines out backwards and upper case.

Example:

Now
Is the
Best time

BEST TIME
IS THE
NOW
import java.util.*;

public class Backwards {
    public static void main(String[] args) {
        String[] lines = new String[50];
        Scanner scan = new Scanner(System.in);
        int pos = 0;
        String t = " "; // initially: t has length = 1
        while (t.length() > 0) {
            t = scan.nextLine();
            lines[pos] = t;
            pos++; // holds next empty location
        }
    }
}
for(int j = pos - 1; j >= 0; j--){
    lines[j] = lines[j].toUpperCase();
    System.out.println(lines[j]);
}
}
processing Now /Is the /Best time

<table>
<thead>
<tr>
<th>null</th>
<th>null</th>
<th>null</th>
<th>null</th>
<th>null</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
processing Now /Is the /Best time

<table>
<thead>
<tr>
<th>Now</th>
<th>Is the</th>
<th>Best time</th>
<th>&quot;&quot;</th>
<th>null</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

final value of pos
Java’s “for-each” construction

Most for loops that apply to arrays march down an entire array of objects, either:

• Collecting information; or

• Altering the contents of objects
public class ArrayTest2{
    public static void main(String args[])
    {
        Infant kid1 = new Infant("a",12);
        Infant kid2 = new Infant("b",12);
        Infant kid3 = new Infant("c",12);
        Infant[] kids = {kid1,kid2,kid3};
    }
}
for (Infant kid : kids) {
    System.out.print(kid.getAge() + " ");
    kid.anotherMonth();
}
System.out.println();

for (Infant kid : kids) {
    System.out.print(kid.getAge() + " ");
}

Prints:
12 12 12
13 13 13
General form

for (Infant kid : kids)

Type tag  variable colon  array
A caveat

You can’t change the array (directly)

- int[] nums = {5,5,5,5,5};

- for(int i : nums)System.out.print(i);
  55555

  for(int i : nums)i++;  // increase nums(?)

  > for(int i : nums)System.out.print(i);
  55555>
public boolean anyBabies(Infant[] kiddo) {
    // are any kids less than 2 months old?
    boolean aBaby = false;
    for (Infant k : kiddo) {
        if (k.getAge() < 2) {
            aBaby = true;
            break;
        }
    }
    return aBaby;
}
boolean majorityOld(Infant[] kiddo, int a) {
    // are strict majority in array older than a?
    int old = 0;
    for (Infant k : kiddo) {
        if (k.getAge() > a) {
            old++;
        } else old--;
    }
    return (old > 0);
}
public String oldest(Infant[] kiddo) {
    if (kiddo.length == 0) return "no kids";
    Infant oldKid = kiddo[0];
    for (Infant k : kiddo) {
        if (k.getAge() > oldKid.getAge()) oldKid = k;
    }
    return(oldKid.getName());
}
What’s wrong with arrays?

1) Size is fixed once array has been created (as in the Backwards example)

2) Elements have fixed positions
How about this problem

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>now</td>
<td>is</td>
</tr>
<tr>
<td>is</td>
<td>the</td>
</tr>
<tr>
<td>the</td>
<td>time</td>
</tr>
<tr>
<td>time</td>
<td>now</td>
</tr>
</tbody>
</table>
Another data structure - an alternate to arrays

The ArrayList

(a trapdoor in the text)
import java.util.*;
public class Backwards2{
public static void main(String[] args){
  ArrayList<String> lines = new ArrayList<String>();
  Scanner scan = new Scanner(System.in);
  int pos = 0; String t = " "; String phrase;
  while(t.length() > 0){
    t = scan.nextLine(); lines.add(t);
  }
  for(int j = lines.size()-1; j >= 0; j--){
    phrase = (lines.get(j)).toUpperCase();
    System.out.println(phrase);
  }
}
### Important operations on `ArrayList` objects

<table>
<thead>
<tr>
<th>Method</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>add(e)</code></td>
<td>appends new element <code>e</code> as last cell</td>
</tr>
<tr>
<td><code>remove(i)</code></td>
<td>removes element at location <code>i</code></td>
</tr>
<tr>
<td><code>set(i,e)</code></td>
<td>sets element at position <code>i</code> to <code>e</code></td>
</tr>
<tr>
<td><code>get(i)</code></td>
<td>returns element at position <code>i</code> in the <code>ArrayList</code></td>
</tr>
<tr>
<td><code>size()</code></td>
<td>returns number of elements in <code>ArrayList</code></td>
</tr>
</tbody>
</table>
The array rotation problem, with an ArrayList
Solution good but not great

```java
String temp = lines.get(0);
lines.remove(0);    // moves everybody up 1
lines.add(temp);    // puts on the end
```

[still, under the hood, plenty of computation!]
Program Design

Most naïve level:

• Input data
• Process data
• Report information

Informed by

• Class structure / how does it fit with decomposition above?

• Object transition analysis / how do objects evolve during program execution

• Method elaboration
Problem #1

The prefix problem - write a program that reads in two strings, then reports if first string is a prefix of second

(EX: hill is prefix of hill-top)

Input / process / report - model

Tip #1 - tell the story

Tip #2 - look for special / extreme cases
The story:

get strings / traverse and judge / report
reporting happens all at once, at the end
Special cases: Str1 bigger than Str2; Str1 empty

Class structure

Scanner for input, inside driver / processing, report also in driver
Problem #2

Given a string, report all instances where 3 consecutive characters are in alphabetical order. (Let the chips fall, in case where capitals, other symbols occur)

So if one, two, three are 3 consecutive symbols, our condition will be:

one <= two && two <= three

Input: a string;
Process & report; check all triples
Special problem - fall off the end..

Input - again, Scanner, in driver

Processing, output - mixed up, and done in a “principal” class, LetterCheck

In driver: read in the String using Scanner, nextLine() method

Make a LetterCheck object, passed the input line

Then process the line
public class LetterCheck{
    private String str;
    private char one = ' ';
    private char two = ' ';
    private char three = ' ';
    private int pos3 = 2;

    public LetterCheck (String s)
    {
        //tricky - load up one, two, three, if s long enough
    }
public void processLine(){
    while (pos3 < str.length()){
        judgeAndReport();
        shiftChars();
    }
}
public void judgeAndReport()
{
    if (judgeInOrder()) report();
}

public boolean judgeInOrder(){ ....... }

public void report(){ ........ }

public void shiftChars(){ ........ }
    // a little tricky - watch for cliff
More on helpers...

Programming Assignment 5 - type in any number of lines of text, end with 2 CRs. The report letter frequencies in the text.

Tip #7 - always look for reuseable idioms
(e.g., use an array as a scoreboard to track frequencies;
look around for the snippit that reads lines of text repeatedly, until 2 CRs)
How do I make an array indexed by letters?

Two helper methods

```c
int myPos(char c){ // assume c is a lower case letter
    return (c - 'a');
}
```

```c
char myChar(int k){ // 0 - 25 map to corr. letter
    return ((char)(k + 'a'));
}
```
Problem #5

Diffy

Create the simulation of this game:

\[
\begin{align*}
\rightarrow & \quad (3 \ 7 \ 10 \ 2) \\
& \quad (4 \ 3 \ 8 \ 1) \\
& \quad (1 \ 5 \ 7 \ 3) \\
& \quad (1 \ 1 \ 2 \ 2) \\
& \quad (0 \ 1 \ 0 \ 1) \\
& \quad (1 \ 1 \ 1 \ 1) \\
& \quad (0 \ 0 \ 0 \ 0)
\end{align*}
\]
A diffy object is a quadruple

In the solution you develop, a sequence of objects evolve..
Problem #6

anay englishay otay igpay atinlay anslatortay

Read in lines (input)

Turn each line into pig latin - a print out (output)
Tip #1 - tell the story
Tip #2 - identify participating classes
Tip #3 - identify problem states
Tip #4 - look for special / extreme cases
Tip #5 - identify actions requiring real attention
Tip #6 - start with a simpler problem
Tip #7 - factor into pieces, use simple helpers
Tip #8 - always look for reuseable idioms
Story: walk through each line, get each word, convert to pig latin, then print out.

(what’s the pig latin rule?)

Overall structure:

  Scanner - driver - PL class

State of PL object: a line of text

Extreme case: word of length 1??

How do I actually pig-latin-ize?
A useful simpler problem - pull words off, maybe print with *’s in between(?)

How to factor

    the key: write a method that’s passed a String (a word, really) and returns its piglatin form

Also - how about helpers: tail(word), isVowel(ch)

Idioms - the “read in multiple lines” idiom.