Manipulating Arrays
Ch 7
Announcements

Midterm scores a bit low...
Highlights

- Partially filled arrays
  (Half full or half empty?)

- for each

```java
double[] grades = new double[20];
for( double d : grades)
```

- Vararg!

```java
public void varargMethod(int... x)
{ }
```
main(String[] args)

When running your Java program, any words from the command line are put into String[] args in main()

```
java HelloWorld hi 2.3456 bye
```

Class name   args[0]   args[1]   args[2]

In NetBeans: Run -> Set Project Config.
-> Customize -> Arguments

(See: MainArgs.java)
<table>
<thead>
<tr>
<th><strong>Optimist</strong></th>
<th><strong>Pessimist</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;The glass is half-full.&quot;</td>
<td>&quot;The glass is half-empty.&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Realist</strong></th>
<th><strong>Idealist</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Yep. That's a glass, alright.&quot;</td>
<td>&quot;One day, cold-fusion from a glass of water will provide unlimited energy and end war.&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Capitalist</strong></th>
<th><strong>Communist</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;If I bottled this and gave it a New Agey sounding name, I could make a fortune.&quot;</td>
<td>&quot;This drink belongs to every single one of us in equal measure.&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Conspiracist</strong></th>
<th><strong>Sexist</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;The government is fluoridating the water for mind-control purposes.&quot;</td>
<td>&quot;This glass isn't gonna refill itself, honeybun...&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Nihilist</strong></th>
<th><strong>Opportunist</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;The glass does not exist, and neither do I.&quot;</td>
<td>&quot;There's a funny t-shirt in here somewhere.&quot;</td>
</tr>
</tbody>
</table>
Partially filled arrays

Arrays are annoying since you cannot change their size

You can get around this by making the array much larger than you need

If you do this you need to keep track of how much of the array you are actually using

(See: ClassList.java)
for each

We often want to run over all elements in an array.

There is a special for-loop that lets you do this in a different way:

```java
int[] x = new int[20];
for (int a : x) {
    // a has the value of x[i] for each i
}
```

(See: ForEach.java)
vararg!
Can make methods that get a variable number of arguments

All the arguments are put into an array

```java
public void varargMethod(int... x)
{
    // The “...” shows it is vararg
    // x is array
}
```

(See: AddOverload.java)
vararg!

You cannot put any arguments after vararg.

You can put arguments before (even of the same type)

```
public void varargMethod(int a, int... x) {
    first int is a second, third, fourth... ints are in x
}
```

(See: VarArg.java)
enum

enum is short for “Enumeration”

This allows you to make an ordered list

Can be useful if you want to associate words with numbers.

(See: Enum.java)
Let's practice arrays by sorting!

Plan of attack:

1. Make a new array
2. Find minimum element in original array and copy into new array
3. Replace minimum element in original array with the maximum element
4. Repeat 2 until done.

(See: Sort.java)