Up to this point, we've created and used one (or a few) objects at a time. Sometimes, we need to create and use many objects of a particular type. Creating and referring to each object separately is tedious and error-prone! Solution: use an array.

Arrays

- Allow for storage of data in one or more dimensions
  - lists
  - tables
  - more complicated relationships (3+ dimensions)
- Can be used to store any type of object

Creating an array in Java

- One-dimension:
  \<data type>\ [] = new <data type>[<size>]

- Multiple-dimension:
  \<data type>\ [] [] ... = new <data type>[<size>][<size>]...
Example: Average high temperature by month in Northfield

- Need 12 “cells” to store temperatures
- Cells will be indexed from 0 to 11
- Can store as integers or doubles
  - we'll store as doubles

Notes

- We can operate on each cell of the array as if it's just a regular variable
- To find the length of an array, use <array name>.length
  - length is a value, not a method!
- Can also initialize array when we declare it:
  - double[] temps = {
    22,29,41,59,72,80,84,81,73,60,41,27
  };
Inserting an item in the middle of an array: example

```java
int [] numbers = { 2, 4, 6, 8, 10 };
int numToInsert = 7;
for (int i=0; i<numbers.length; i++) {
    if (numbers[i] > numToInsert) {
        break;
    }
}
int index = i; // this is where we will insert 7
// need to expand the array
int [] temp = new int[numbers.length + 1];
for (int j=0; j<numbers.length + 1; j++) {
    if (j < index)
        temp[j] = numbers[j];
    else if (j == index)
        temp[j] = numToInsert;
    else
        temp[j] = numbers[j-1];
}
numbers = temp;
```

Java's built-in array classes (in `java.util`)

- **ArrayList**
- **Vector**
- Both of these inherit from Java's List and Collections interfaces

**ArrayList**

- Start with a default capacity
- As items are added, capacity is automatically increased
- Can add or remove items from anywhere in the array
  - will automatically be resized
- Can set/reset items at any position in the array
- Can fill with any type of object
  - generically add `Object` items to the array
  - need to make sure our items are subclasses of `Object`!

Example: Inserting an item into an ArrayList

```java
import java.util.*;
...
ArrayList numbers = new ArrayList();
temp = numbers.get(index)
int numToInsert = 7;
for (int i=0; i<numbers.size(); i++) {
    if (((Integer)(numbers.get(i))).intValue() > numToInsert) {
        index = i; // this is where we will insert 7
        break;
    }
}
numbers.add(index, new Integer(numToInsert));
```