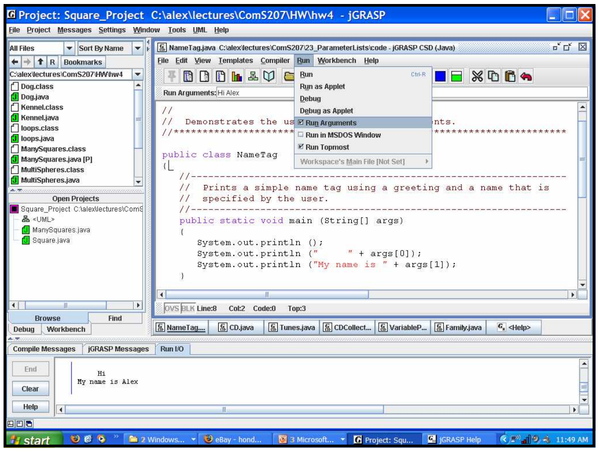
2D Arrays

October 13, 2006

ComS 207: Programming I (in Java)
Iowa State University, FALL 2006
Instructor: Alexander Stoytchev

Quick review of last lecture



Run Arguments: Hi Alex

```
//
//  Demonstrates the us                    nts.
//*******************                      ***********************

public class NameTag
{
   //-----------------------------------------------------------------
   //  Prints a simple name tag using a greeting and a name that is
   //  specified by the user.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      System.out.println ();
      System.out.println ("      " + args[0]);
      System.out.println ("My name is " + args[1]);
   }
```

```
      Hi
My name is Alex
```

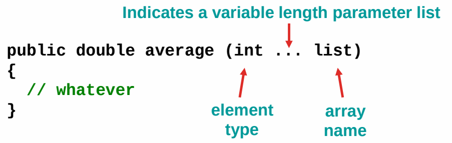
Variable Length Parameter Lists

- Suppose we wanted to create a method that processed a different amount of data from one invocation to the next
- For example, let's define a method called `average` that returns the average of a set of integer parameters

```
// one call to average three values
mean1 = average (42, 69, 37);

// another call to average seven values
mean2 = average (35, 43, 93, 23, 40, 21, 75);
```

Variable Length Parameter Lists

- Using special syntax in the formal parameter list, we can define a method to accept any number of parameters of the same type
- For each call, the parameters are automatically put into an array for easy processing in the method


Indicates a variable length parameter list

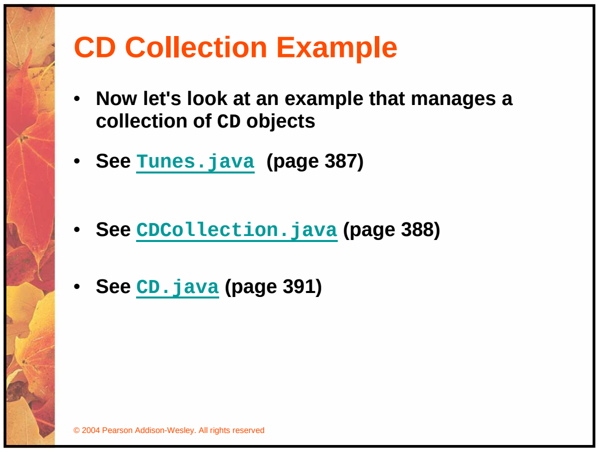
```
public double average (int ... list)
{
  // whatever
}
```

element type

array name

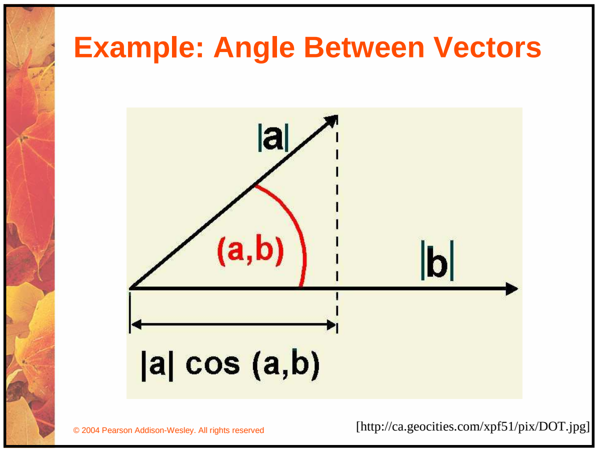
Variable Length Parameter Lists

```
public double average (int ... list)
{
   double result = 0.0;

   if (list.length != 0)
   {
      int sum = 0;
      for (int num : list)
         sum += num;
      result = (double)sum / list.length;
   }

   return result;
}
```



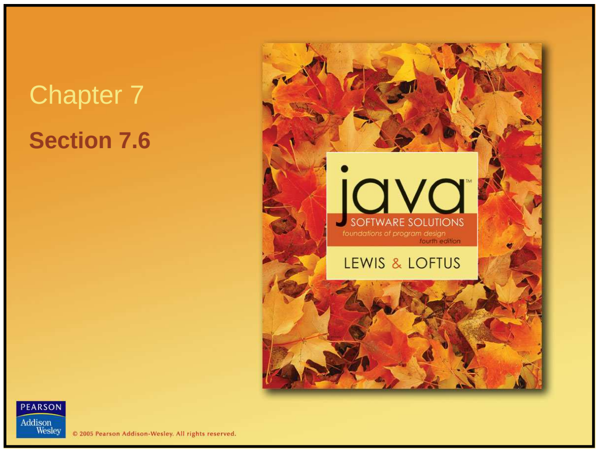
CD Collection Example

- Now let's look at an example that manages a collection of `CD` objects
- See `Tunes.java` (page 387)
- See `CDCollection.java` (page 388)
- See `CD.java` (page 391)



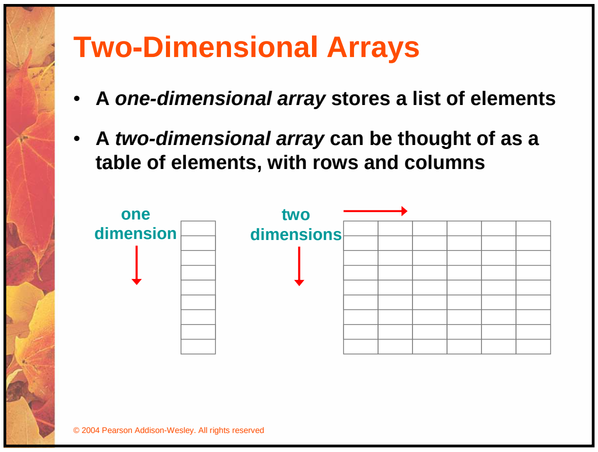
Example: Angle Between Vectors

[http://ca.geocities.com/xpf51/pix/DOT.jpg]



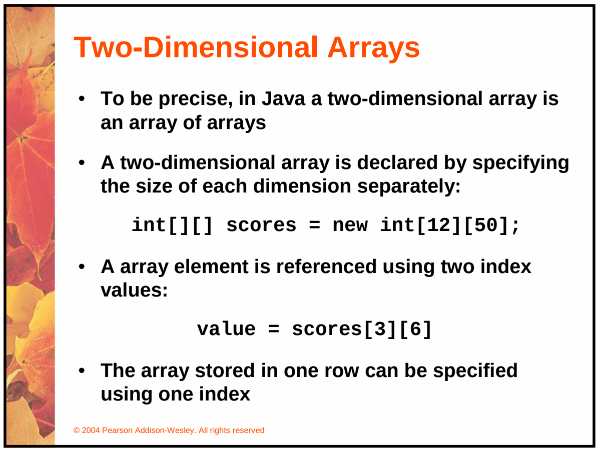
Chapter 7

Section 7.6



Two-Dimensional Arrays

- A *one-dimensional array* stores a list of elements
- A *two-dimensional array* can be thought of as a table of elements, with rows and columns



Two-Dimensional Arrays

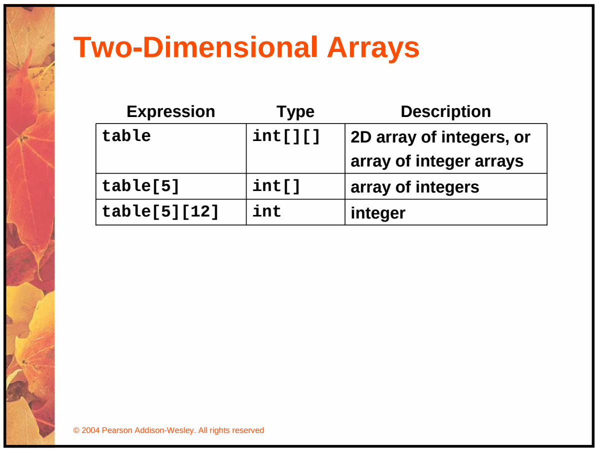
- To be precise, in Java a two-dimensional array is an array of arrays
- A two-dimensional array is declared by specifying the size of each dimension separately:

```
int[][] scores = new int[12][50];
```

- A array element is referenced using two index values:

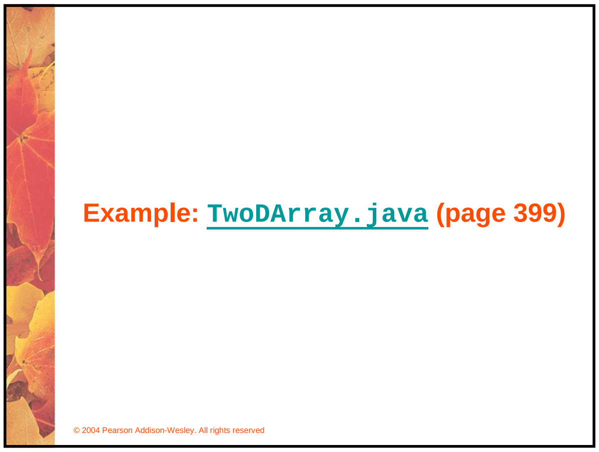
```
value = scores[3][6]
```

- The array stored in one row can be specified using one index

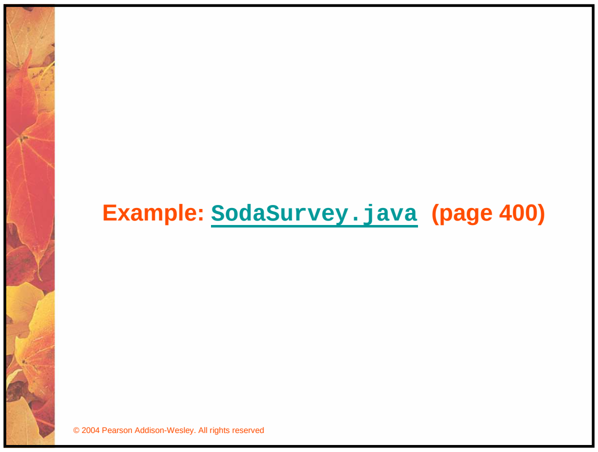


Two-Dimensional Arrays

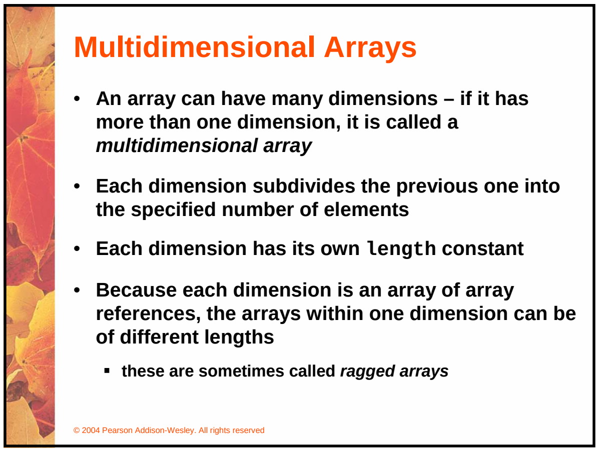
Expression	Type	Description
`table`	`int[][]`	2D array of integers, or array of integer arrays
`table[5]`	`int[]`	array of integers
`table[5][12]`	`int`	integer



Example: `TwoDArray.java` (page 399)

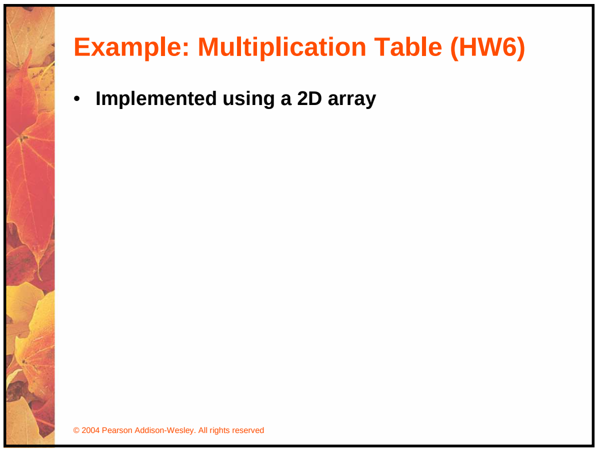


Example: `SodaSurvey.java` (page 400)



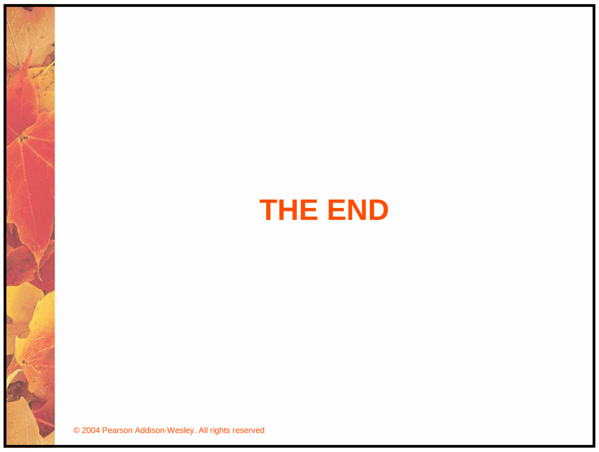
Multidimensional Arrays

- An array can have many dimensions – if it has more than one dimension, it is called a ***multidimensional array***
- Each dimension subdivides the previous one into the specified number of elements
- Each dimension has its own `length` constant
- Because each dimension is an array of array references, the arrays within one dimension can be of different lengths
  - these are sometimes called *ragged arrays*



Example: Multiplication Table (HW6)

- Implemented using a 2D array



THE END