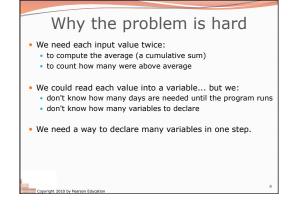
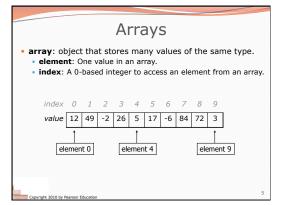
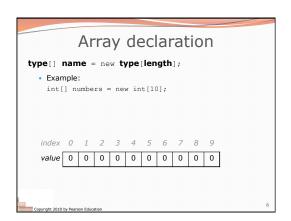
Building Java Programs Chapter 7 Lecture 7-1: Arrays reading: 7.1

Receipts from file • Write a program that prints out the receipts for many meals. Input file (meals.txt): 5.85 2.85 3.95 10.95 11 7.99 8.99 10 4.50 5.50 22 21.75 24 8 5.50 1.50 3.99 17.00 Output: Table 1: Subtotal: \$34.60; Tax: \$2.77; Tip: \$5.19; Total: \$42.56 Table 2: Subtotal: \$118.23; Tax: \$9.46; Tip: \$17.73; Total: \$145.42 Table 3: Subtotal: \$22.49; Tax: \$1.80; Tip: \$3.37; Total: \$27.66

Can we solve this problem? • Consider the following program (input underlined): How many days' temperatures? 7 Day 1's high temp: 45 Day 2's high temp: 44 Day 3's high temp: 39 Day 4's high temp: 39 Day 6's high temp: 37 Day 6's high temp: 46 Day 7's high temp: 46 Day 7's high temp: 53 Average temp = 44.6 4 days were above average.







Array declaration, cont.

• The length can be any integer expression.

```
int x = 2 * 3 + 1;
int[] data = new int[x % 5 + 2];
```

• Each element initially gets a "zero-equivalent" value.

Туре	Default value
int	0
double	0.0
boolean	false
String or other object	null (means, "no object")

Copyright 2010 by Pearson Education

Accessing array elements

Arrays of other types

```
double[] results = new double[5];
results[2] = 3.4;
results[4] = -0.5;

index 0 1 2 3 4

value 0.0 0.0 3.4 0.0 -0.5

boolean[] tests = new boolean[6];
tests[3] = true;

index 0 1 2 3 4 5

value false false false true false false
```

Out-of-bounds

- Legal indexes: between 0 and the array's length 1.
- Reading or writing any index outside this range will throw an ArrayIndexOutOfBoundsException.
- Example

value 0 0 0 0 0 0 0 0 0 0

Copyright 2010 by Pearson Education

Arrays and for loops

• It is common to use for loops to access array elements.

```
for (int i = 0; i < 8; i++) {
    System.out.print(numbers[i] + " ");
}
System.out.println(); // output: 0 4 11 0 44 0 0 2</pre>
```

• Sometimes we assign each element a value in a loop.

```
for (int i = 0; i < 8; i++) {
    numbers[i] = 2 * i;
}

index 0 1 2 3 4 5 6 7

value 0 2 4 6 8 10 12 14
```

Copyright 2010 by Pearson Education

The length field • An array's length field stores its number of elements. name.length for (int i = 0; i < numbers.length; i++) { System.out.print(numbers[i] + " "); } // output: 0 2 4 6 8 10 12 14 • It does not use parentheses like a String's .length(). • What expressions refer to: • The last element of any array? • The middle element?

```
Weather question

• Use an array to solve the weather problem:

How many days' temperatures? 7

Day 1's high temp: 45

Day 2's high temp: 44

Day 3's high temp: 39

Day 4's high temp: 48

Day 5's high temp: 37

Day 6's high temp: 46

Day 7's high temp: 53

Average temp = 44.6

4 days were above average.
```

```
Quick array initialization

type[] name = {value, value, ... value};

• Example:
int[] numbers = {12, 49, -2, 26, 5, 17, -6};
index 0 1 2 3 4 5 6
value 12 49 -2 26 5 17 -6

• Useful when you know what the array's elements will be
• The compiler figures out the size by counting the values
```

```
"Array mystery" problem

• traversal: An examination of each element of an array.

• What element values are stored in the following array?

int[] a = {1, 7, 5, 6, 4, 14, 11};
for (int i = 0; i < a.length - 1; i++) {
    if (a[i] > a[i + 1]) {
        a[i + 1] = a[i + 1] * 2;
    }

index 0 1 2 3 4 5 6

value 1 7 10 12 8 14 22
```

The Arrays class

• Class Arrays in package java.util has useful static methods for manipulating arrays:

Method name	Description
binarySearch(array, value)	returns the index of the given value in a sorted array (or < 0 if not found)
copyOf (array, length)	returns a new copy of an array
equals(array1, array2)	returns true if the two arrays contain same elements in the same order
fill(array, value)	sets every element to the given value
sort (array)	arranges the elements into sorted order
toString(array)	returns a string representing the array, such as "[10, 30, -25, 17]"

• Syntax: Arrays.methodName(parameters)

```
Arrays.toString
• Arrays.toString accepts an array as a parameter and returns a String representation of its elements.
     int[] e = {0, 2, 4, 6, 8};
     e[1] = e[3] + e[4];
    System.out.println("e is " + Arrays.toString(e));
    e is [0, 14, 4, 6, 8]
  • Must import java.util.*;
```

Weather question 2

• Modify the weather program to print the following output:

```
How many days' temperatures? 7
Day 1's high temp: 45
Day 2's high temp: 44
Day 3's high temp: 39
Day 4's high temp: 39
Day 4's high temp: 37
Day 6's high temp: 37
Day 6's high temp: 46
Day 7's high temp: 53
Average temp = 44.6
4 days were above average.
   Temperatures: [45, 44, 39, 48, 37, 46, 53]
Two coldest days: 37, 39
Two hottest days: 53, 48
```

```
Weather answer 2
// Reads temperatures from the user, computes average and \sharp days above average import java.util.*;
public class Weather2 {
   public static void main(String[] args) {
        int[] temps = new int[days];
... (same as Weather program)
                                               // array to store days' ter
        // report results
System.out.printf("Average temp = %.1f\n", average);
System.out.println(count + " days above average");
```