Introduction to Arrays

CSC 1051 – Data Structures and Algorithms I
Dr. Mary-Angela Papalaskari
Department of Computing Sciences
Villanova University

Course website: www.csc.villanova.edu/~map/1051/

Some slides in this presentation are adapted from the slides accompanying Java Software Solutions by Lewis & Loftus

Arrays

• An array is an ordered list of values

```
0  1  2  3  4  5  6  7  8  9
7.9 8.7 9.4 8.2 6.7 9.8 8.7 8.1 7.4 9.1
```

This array holds 10 values of type double, indexed by 0, 1, ..., 9

Implementing arrays in Java:
• Declaration of an array
• Instantiation of the object that represents the array
• Initialization of the array values

Arrays - Declaration

Declaration: double[] scores;

The entire array has a single name

scores 0 1 2 3 4 5 6 7 ...

Arrays - Instantiation

Declaration: double[] scores = new double[10];

Instantiation:

scores 0 1 2 3 4 5 6 7 8 9

Size of array

scores.length 10
Arrays - Initialization

Declaration:
double[] scores = new double[10];

Instantiation:

Initialization:

<table>
<thead>
<tr>
<th>scores</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.9</td>
<td>8.7</td>
<td>9.4</td>
<td>8.2</td>
<td>6.7</td>
<td>9.8</td>
<td>8.7</td>
<td>8.1</td>
<td>7.4</td>
<td>9.1</td>
</tr>
</tbody>
</table>

Size of array: scores.length = 10

Using Arrays

Array elements can be assigned a value, printed, or used in a calculation. Examples:

```java
System.out.println("Top = " + scores[5]);
mean = (scores[0] + scores[1])/2;
scores[3] = 7 + Math.random();
scores[scores.length - 1] = 9.0;
double num = scores[rand.nextInt(10)];
```

Declaring and instantiating Arrays

• More examples:

```java
int[] weights = new int[2000];
boolean[] flags;
flags = new boolean[20];
char[] codes = new char[1750];
double[] prices = new double[500];
```

Try this: Write some Java code to create an array

• declare and instantiate an array ratings that holds 5 values type int

```java
ratings
```

• declare and instantiate an array vowel to hold 5 values of type char, then initialize its values to the vowels 'a', 'e', 'i', 'o', 'u'

```java
vowel
```

CSC 1051 M.A. Papalaskari, Villanova University
What gets printed?

```java
System.out.println (scores[8] + 1);
System.out.println (scores[1] + scores[2]);
System.out.println (scores[1 + 2]);
System.out.println(scores[scores.length - 2]);
```

Show how `scores` values change:

```java
scores[4] = 1;
scores[5] = scores[0] + 1;
scores[scores.length - 2]) = 5.5;
```

Processing Arrays using for-loops:
1) draw a picture of the resulting array

```java
double[] mylist = new double[10];
for (int i = 0; i < 10; i++)
    mylist[i] = 0;

for (int i = 0; i < 10; i++)
    mylist[i] = i;
```

Processing Arrays using for-loops:
2) Reversing through...

```java
double[] tinyScores = new double[5];
for (int i = 4; i >= 0; i--)
    System.out.println(tinyScores[i]);
```

Output:
Processing Arrays using for-loops:

3) write a for-loop to print the values in the vowel array (going forward)

4) write a for-loop to print the values in the vowel array (going backward)

Bounds Checking

An array index must specify a valid element

- Example: If an array codes holds 100 values, it can be indexed from 0 to 99. If the value of count is 100, then
  System.out.println(codes[count]);
  causes an ArrayIndexOutOfBoundsException

- It’s common to introduce off-by-one errors when using arrays:
  for (int index=0; index <= 100; index++)
  codes[index] = index*50 + epsilon;

Initializer Lists

- Alternative way to declare, instantiate, and initialize an array. For example:
  int[] ratings = {4, 3, 3, 1, 4, 2, 1, 0, 3, 4};
  char[] grades = {'A', 'B', 'C', 'D', 'F'};

- NOTE:
  - the new operator is not used
  - size of array is determined by the number of items listed
  - can only be used in the array declaration

The “for-each” Loop

- A simple way of processing every array element:
  for (double score : scores)
    System.out.println (score);

- NOTE:
  - Only appropriate when processing all array elements starting at index 0
  - It can’t be used to set the array values
Another example

String[] animals = {"dog", "cat", "mouse", "fox"};
for (String word : animals)
    System.out.println("The " + word + " ate the cake");

for (String word : animals)
    for (String otherWord : animals)
        System.out.println("The " + word + " ate the " + otherWord);

Try this: Use the “for each” loop to scan through an array of int containing ratings (range: 0 - 4) and count up how many 4’s.

int[] ratings = {4, 3, 3, 1, 4, 3, 1, 0, 3, 4};

Try this: Repeat, but now count up the 0’s, 1’s,… 4’s – Use a separate array for this

int[] ratings = {4, 3, 3, 1, 4, 3, 1, 0, 3, 4};

More array examples (see textbook):
- BasicArray.java
- Primes.java
- ReverseOrder.java
- LetterCount.java
Another example: Computing letter frequency counts

Enter a sentence:
In Casablanca, Humphrey Bogart never says "Play it again, Sam."

A: 0    a: 10
B: 1    b: 1
C: 1    c: 1
D: 0    d: 0
E: 0    e: 3

Let's write a program to do this

```java
// LetterCount.java  Author: Lewis/Loftus
// Demonstrates the relationship between arrays and strings.
import java.util.Scanner;
public class LetterCount {
  //////////////////////////////////////////////////////////////////
  //  Reads a sentence from the user and counts the number of
  //  uppercase and lowercase letters contained in it.
  //////////////////////////////////////////////////////////////////
  public static void main (String[] args)
  {
    final int NUMCHARS = 26;
    Scanner scan = new Scanner (System.in);
    int[] upper = new int[NUMCHARS];
    int[] lower = new int[NUMCHARS];
    char current; // the current character being processed
    int other = 0; // counter for non-alphabetics
    continue
    System.out.println ("Enter a sentence: ");
    String line = scan.nextLine();
    // Count the number of each letter occurrence
    for (int ch = 0; ch < line.length(); ch++)
    {
      current = line.charAt(ch);
      if (current >= 'A' && current <= 'Z')
        upper[current-'A']++;  
      else
        if (current >= 'a' && current <= 'z')
          lower[current-'a']++;  
        else
          other++;  
    }  
    // Print the results
    System.out.println ("Enter a sentence: ");
    for (int letter=0; letter < upper.length; letter++)
    {
      System.out.print ( (char) (letter + 'A') );
      System.out.print (" ");
      System.out.print (" ");
    }  
    System.out.println ();
    System.out.println ("Non-alphabetic characters: ");
    continue
  }  
}
```