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/

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 that are indexed from 0 to 9

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];
```

The entire array has a single name.

```
scores
0 1 2 3 4 5 6 7 8 9
```

```
scores[2]
```

Index

```
scores.length
10
```

Size of array

```
```
Arrays - Initialization

**Declaration:**
The entire array has a single name.

```java
double[] scores = new double[10];
```

**Instantiation:**

**Index**

<table>
<thead>
<tr>
<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>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>

**Element Type**

```
scores[2]
```

**Size of Array**

```
scores.length = 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];
  ```

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)];
```

Try this: Write some Java code to create an array

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

- 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’
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 = new double[]{7.9, 8.7, 9.4, 8.2, 6.7, 9.8, 8.7, 8.1, 7.4, 9.1};
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
  ```java
  System.out.println(codes[count]);
  ```
  causes an `ArrayIndexOutOfBoundsException`

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

CSC 1051 Villanova University