Intro to Arrays, continued

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 - Review

• **Declaration:**

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

The entire array has a single name

- The array holds 10 values of type `double`
- Indexed from 0 to 9
- Size given by `scores.length = 10`

• **Initialization:**

```java
scores[0] = 7.9;
scores[1] = 8.7;
scores[2] = 9.4;
scores[3] = 8.2;
scores[4] = 6.7;
scores[5] = 9.8;
scores[6] = 8.7;
scores[7] = 8.1;
scores[8] = 7.4;
scores[9] = 9.1;
```

- This array holds 10 values of type `double`
- Indexed from 0 to 9
- Size given by `scores.length = 10`
Initializer Lists

• Alternative way to declare, instantiate, and initialize an array. For example:

```java
int[] units = {14, 32, 89, 9, 54, 30};
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
Making a copy of an array

• Suppose we have two arrays:

```java
int[] a = {147, 323, 89, 933};
int[] b = {100, 200, 300, 400};
```

```java
for (int i = 0; i < a.length; i++)
    a[i] = b[i];
```
Making a copy of an array

• Suppose we have two arrays:
  ```
  int[] a = {147, 323, 89, 933};
  int[] b = {100, 200, 300, 400};
  ```

• Compare the following:
  ```
  for (int i=0; i<a.length; i++)
      a[i] = b[i];
  ```

CSC 1051 M.A. Papalaskari, Villanova University
Arrays are **Objects**

- What happens when we do this?

  ```
  int[] a = {147, 323, 89, 933};
  int[] b = {100, 200, 300, 400};
  a = b;
  ```

- Or this?

  ```
  int[] c = {100, 200};
  a = c;
  ```
Arrays as Parameters

• An entire array can be passed as a parameter to a method (just like any other object). For example:

```java
int[] a = {1, 2, 3, 4};
doSomething(a);
```

• Assumes a definition for `doSomething()`, for example:

```java
static void doSomething(int[] x)
{
    x[0] = 5;
}
```
Arrays as Parameters

• An entire array can be passed as a parameter to a method (just like any other object).

• **Exercise:** Write a method that adds 2 to the value of each element in an array of type `double[]`. 
Command-Line Arguments

• It turns out we have been using arrays as parameters all along!

public static void main (String[] args)
Command-Line Arguments

• It turns out we have been using arrays as parameters all along!

```java
public class Test {
    public static void main (String[] args)
    {
        System.out.println ();
        System.out.println (" " + args[0]);
        System.out.println (" " + args[1]);
    }
}
```

• These values come from *command-line arguments* that are provided when the interpreter is invoked

• jGrasp calls them “Run Arguments”
More array examples (see textbook):

- `BasicArray.java`
- `Primes.java`
- `ReverseOrder.java`
- `LetterCount.java`
Another example of the use of arrays: Computing letter frequency counts

Sample run:

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
Letter Frequency counts - Algorithm

Uses two arrays: upper and lower with 26 elements each, to count the number of upper- and lowercase letters, respectively.

1. declare and instantiate upper and lower with 26 elements of type int
2. prompt user to "Enter a sentence:” and scan input into a String line
3. for each character current in line do the following:
   – if (current is between ‘A’ and 'Z')
     • position ← current - ‘A’  // compute distance from ‘A’
     • upper[position] ← upper[position] + 1
       // increment count for current
   – else if (current is between ‘a’ and ‘z')
     • position ← current – ‘a’
     • lower[position] ← upper[position] + 1
   – else do nothing  // current is not a letter
import java.util.Scanner;

public class LetterCount {
    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++;
}

continue
// Print the results
System.out.println();
for (int letter=0; letter < upper.length; letter++)
{
    System.out.print((char)(letter + 'A'));
    System.out.print(" ": " + upper[letter]);
    System.out.print("\t\t" + (char)(letter + 'a'));
    System.out.println(" ": " + lower[letter]);
}

System.out.println();
System.out.println("Non-alphabetic characters: " + other);
**Sample Run**

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
F: 0     f: 0
G: 0     g: 2
H: 1     h: 1
I: 1     i: 2
J: 0     j: 0
K: 0     k: 0
L: 0     l: 2
M: 0     m: 2
N: 0     n: 4
O: 0     o: 1
P: 1     p: 1
Q: 0     q: 0
R: 0     r: 3
S: 1     s: 3
T: 0     t: 2
U: 0     u: 1
V: 0     v: 1
W: 0     w: 0
X: 0     x: 0
Y: 0     y: 3
Z: 0     z: 0

Non-alphabetic characters: 14