Computer Programming Using JAVA
COP 2800 - Fall 2016

Lecture 10: Java Array

By Ahmed E. Khaled
Recap to previous lectures!

- Define constant in java?
- The different methods for Java string?
int \( x = (\text{int})(\text{Math.random()} \times ((10 - 0) + 1) + 0); \)  // random number from 0 to 10 including 10

int \( y = (\text{int})(\text{Math.random()} \times ((10 - 0) \quad + 0); \)  // random number from 0 to 10 excluding 10
Lecture 10 : Java Arrays

- Data structure.
- Java 1D Arrays.
- Common examples.
- Array errors.
Data structure

Data Structure is a way to *organize data* in such a way that it can be used efficiently.

The basic one that it is built into every programming language: the *Array*. 
An array is a data structure:

1. Numbered list of items
2. All items are of same type
3. Numbered from zero up to some maximum value.

For example, an array might contain 10 integers, numbered from zero to 9.
• The number of items in an array is called *the length* of the array.

• The position number of an item in an array is called the *index* of that item.
Define array in Java!

```java
int[] list;
```

- Variable named `list` of type `int[]`, refers to an array of ints, but initially its value is null or undefined.

- The `new` operator is used to create a new array object, which can then be assigned to this variable.

```java
list = new int[5]; //creates an array of 5 integers.
```
int[] mylist = new int[5];

int[] mylist;
mylist = new int[5];
Assign value to an item

mylist[k] = value;

mylist is an array of size 5, so k can be 0,1,2,3 or 4
int[] mylist = new int[5];
mylist[2] = 100;
System.out.println("value of 3rd item is " + mylist[2]);
Every use of a variable in a program specifies a memory location.

To the computer, mylist[k] means something like this:
1) Get the pointer that points to the variable mylist.
2) Follow this pointer to find an array object.
3) Get the value of k.
4) Go to the k-th position in the array, and that’s the memory location you want.
.. then set some values!

```java
int[] list = new int[5];

list[0] = 12;
list[1] = 50;
list[2] = 34;
list[3] = 57;
list[4] = 100;

System.out.println("first index : "+list[0]);
System.out.println("second index : " + list[1]);
System.out.println("third index : " + list[2]);
System.out.println("forth index : " + list[3]);
System.out.println("fifth index : " + list[4]);
```
Array Length

```java
int[] list = new int[5];
System.out.println("size of the array is : "+ list.length);
```

The index for the *last item* is (list.length – 1).

You are not allowed to change length’s value (in this case = 5) after been initialized.
Array Initialization

Another way to initialize array is by assigning values like:

```java
int[] list = { 1, 4, 9, 16, 25, 36, 49 };```

The length of list is seven, since seven values are provided in the initializer.

```java
System.out.println("forth element is " + list[3])
```
In Java, a newly created array is always filled with a known, default value: `zero` for numbers, `false` for boolean, the character with Unicode number zero for char, and `null` for objects.

```java
int[] list = new int[5];

System.out.println("size of the array is : " + list.length);
System.out.println("first index : " + list[0]);
System.out.println("second index : " + list[1]);
System.out.println("third index : " + list[2]);
System.out.println("forth index : " + list[3]);
System.out.println("fifth index : " + list[4]);
```

size of the array is : 5
first index : 0
second index : 0
third index : 0
forth index : 0
fifth index : 0
BUILD SUCCESSFUL (total time: 1 second)
• How to traverse Arrays with Loops and print values?
Linking array with loops

```java
int[] list = new int[5];

System.out.println("size of the array is : "+ list.length);

list[0] = 12;
list[1] = 50;
list[2] = 34;
list[3] = 57;
list[4] = 100;

for(int i=0 ; i<list.length ; i++){
    System.out.println("the value of item number "+ i + " is " + list[i]);
}
```

run:
size of the array is : 5
the value of item number 0 is 12
the value of item number 1 is 50
the value of item number 2 is 34
the value of item number 3 is 57
the value of item number 4 is 100
BUILD SUCCESSFUL (total time: 0 seconds)
Linking array with loops

int count = 0;

while(count < list.length){
    System.out.println("Item with index " + count + " valued " + list[count]);
    count = count + 1;
}

• How to initializes an array named myList with *random values* between 0.0 and 100.0, but less than 100.0?
Initialize array with random values

```java
for (int i = 0; i < myList.length; i++) {
    myList[i] = Math.random() * 100;
}
```
• Suppose, for example, that A is an array of type double and the goal is to *add up all the numbers in the array*?
Linking array with loops

Double[] A = new double[5];
double sum = 0; // Start with 0.
for (int i = 0; i < A.length; i++)
    sum = sum + A[i]; // add A[i] to the sum
}
Type of array errors

If the value of list is null, then list doesn’t even refer to an array. The attempt to refer to an element of an array that doesn’t exist is an error that will cause an exception of type `NullPointerException` to be thrown.

```java
int[] list = null;
System.out.println( list[2] );
```

run:

```
Exception in thread "main" java.lang.NullPointerException
  at welcomejava.WelcomeJava.main(WelcomeJava.java:12)
Java Result: 1
BUILD SUCCESSFUL (total time: 1 second)
```
Type of array errors

For list[k], if k >= list.length. This is called an “array index out of bounds” error.

```java
int[] list = new int[5];
list[0] = 12;
list[1] = 50;
list[2] = 34;
list[3] = 57;
list[4] = 100;
System.out.println( list[5] );
```

```
int[] list = new int[5];
list[0] = 12;
list[1] = 50;
list[2] = 34;
list[3] = 57;
list[4] = 100;
System.out.println( list[6] );
```

run:

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
    at welcomejava.WelcomeJava.main(WelcomeJava.java:16)
Java Result: 1
BUILD SUCCESSFUL (total time: 0 seconds)
Type of array errors

For list[k], if k< 0. This is called an “array index out of bounds” error.

```java
int[] list = new int[5];

list[0] = 12;
list[1] = 50;
list[2] = 34;
list[3] = 57;
list[4] = 100;

System.out.println( list[-1] );
```

run:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
    at welcomejava.WelcomeJava.main(WelcomeJava.java:16)
Java Result: 1
BUILD SUCCESSFUL (total time: 0 seconds)
```
Arrays Examples

- Random Values.
- Search Array.
- Counting.
Random Values

```java
int[] sourceArray = new int[5];

for(int i=0 ; i < 5 ; i++){
    sourceArray[i] = (int) (Math.random() * 10);
}

for(int i=0 ; i < 5 ; i++){
    System.out.print(sourceArray[i] + " ");
}
System.out.println();
```
public static boolean searchArray(int[] myArray, int key) {
    for(int i = 0 ; i < myArray.length ; i++){
        if(myArray[i] == key){
            return true; //return from function
        }
    }
    return false;
}

public static void main(String[] args) {
    int[] sourceArray = new int[5];
    sourceArray[0] = 100;
    sourceArray[1] = 200;
    sourceArray[2] = 300;
    sourceArray[3] = 400;
    sourceArray[4] = 500;

    boolean found = searchArray(sourceArray, 300);
    if(found){
        System.out.println("Found it !");
    }else{
        System.out.println("Key is not Found it !");
    }
}
public static int searchArray(int[] myArray, int key) {
    for(int i = 0 ; i < myArray.length ; i++){
        if(myArray[i] == key){
            return i;         //return from function
        }
    }
    return -1;
}

public static void main(String[] args) {
    int[] sourceArray = new int[5];
    sourceArray[0] = 100;
    sourceArray[1] = 200;
    sourceArray[2] = 300;
    sourceArray[3] = 400;
    sourceArray[4] = 500;

    int index = searchArray(sourceArray, 300);
    if(index != -1){
        System.out.println("Found it at "+ index);
    }else{
        System.out.println("Key is not Found it! ");
    }
}
public static int searchArray(int[] myArray, int key) {
    int totalCount = 0;
    for(int i = 0; i < myArray.length; i++){
        if(myArray[i] == key){
            totalCount = totalCount + 1;
        }
    }
    return totalCount;
}

public static void main(String[] args) {
    int[] sourceArray = new int[5];
    sourceArray[0] = 100;
    sourceArray[1] = 200;
    sourceArray[2] = 300;
    sourceArray[3] = 300;
    sourceArray[4] = 500;
    int count = searchArray(sourceArray,300);
    System.out.println("This key is found "+count+" times");
}