Recap of Primitive Arrays

Produced by: Mairead Meagher
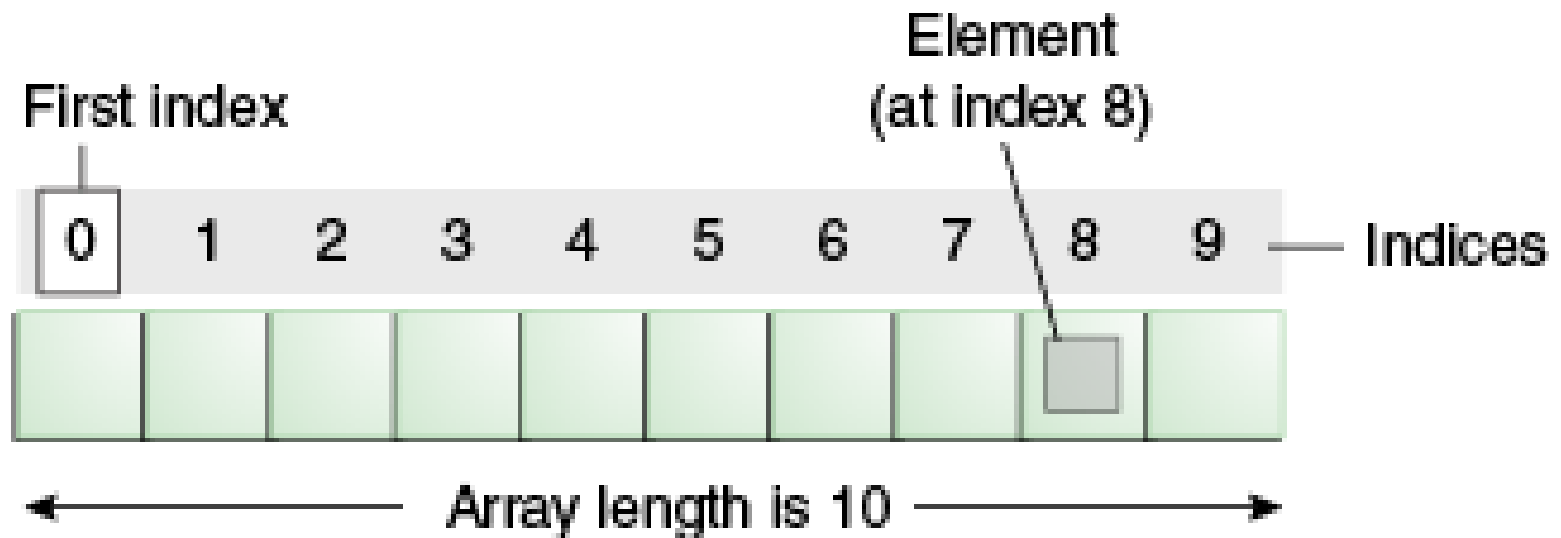   Dr. Siobhán Drohan

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
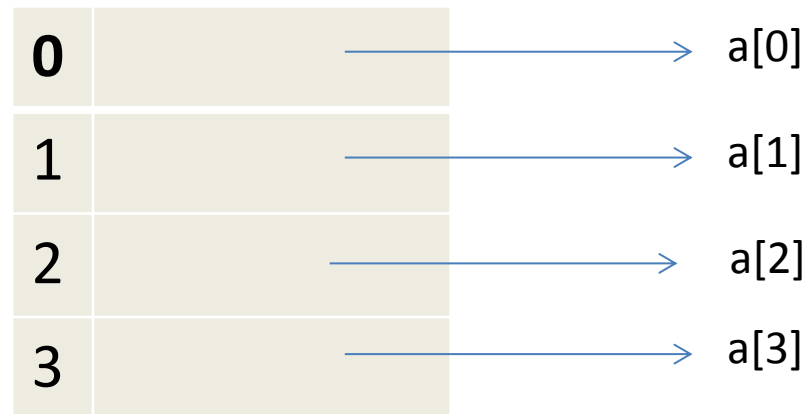http://www.wit.ie/

Arrays (fixed-size collections)

- Arrays are a way to collect associated values.
- Programming languages usually offer a special fixed-size collection type: an *array*.

Example of a primitive array

- This is a box made up of four sub-divisions called 0, 1, 2 and 3.
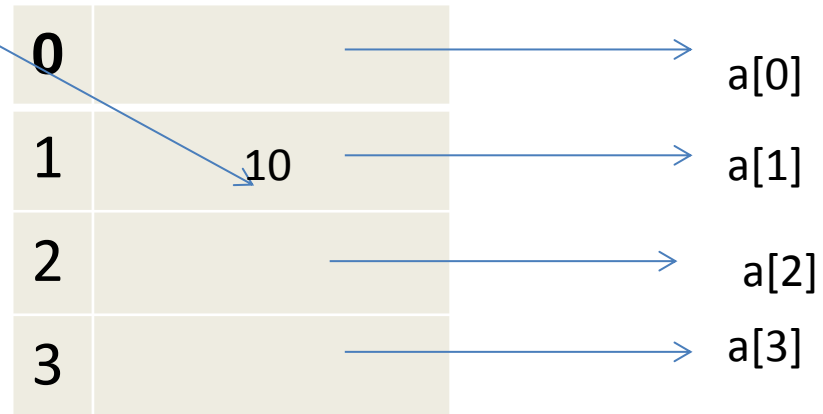
- **NOTE :THE FIRST POSITION IS 0.**

```
int[]  a;
a = new int[4];
```

0	→ a[0]
1	→ a[1]
2	→ a[2]
3	→ a[3]

Accessing elements of an array

- You can access any element separately, e.g.
  a[1] = 10;

0	
1	10
2	
3	

a[0]

a[1]

a[2]

a[3]

Rules for primitive Arrays

1. When you declare an array of a specific type, each sub-section (element) of the array is of the same declared type.

2. The size of the array, i.e. how many sections (elements) in the array is denoted by the number in the square bracket in the following statement:

    int[] a = new int[4];

```
int[]  a;
.
.
.
.
a = new int[4];
```

```
int[]  a =  new int[4];
```

Different ways to declare arrays

```
int[] numbers = { 4, 1, 22, 9};
```

Declaring an Array using literals

declaration and initialisation at the same time

```
private int[] numbers = { 3, 15, 4, 5 };



System.out.println(numbers[1]);
```
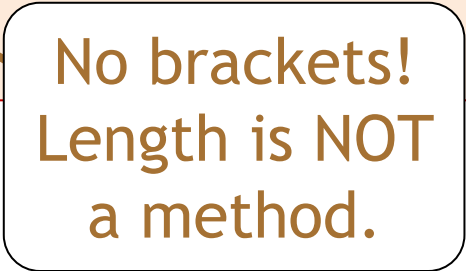
NOTE: literals can only be used when declaring an array.

Array length

```
private int[] numbers = { 3, 15, 4, 5 };



int n = numbers.length;
```

No brackets!
Length is NOT
a method.

```
for(int i = 0; i < numbers.length; i++)
{
    System.out.println(i + ": " + numbers[i]);
}
```

What types can be stored in arrays?

- An array can store any type of data, either:
  - Object types or
  - Primitive types

int a[] = new int[10];

String words = new String[30];

Spot spots[] = new Spot[20];

```java
import javax.swing.*;

int a[];
int numData =
        Integer.parseInt(JOptionPane.showInputDialog(
        "How many values do you wish to sum? ", "3"));

//now, use this value to make the array this size
a = new int[numData];
int sum = 0;
for (int i = 0; i < numData ; i ++)  {
        a[i] = Integer.parseInt(JOptionPane.showInputDialog(
                "Please enter a number ", "3"));
        sum += a[i];
 }

 println("The sum of the values you typed in is : " + sum);
```

If we wanted to change how many numbers we want to add…

Do we have to use all elements in the array?

- No. We may not know how many elements of the array will actually be used e.g.
  - We wish to store an average mark for each of the 50 students in a particular class → create an array of 50 elements.
  - However, not all students might have sat their assessments; perhaps only 45 did → only 45 of the elements will be populated with an average mark.

Do we have to use all elements in the array?

- When not all elements in an array are populated, we need to:
  - have another variable (e.g. int size) which contains the number of elements of the array is actually used.
  - ensure size is used when processing the array e.g.

    for (int i= 0; i < size; i++)

Questions?

Waterford Institute *of* Technology

INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/