

Primitive Arrays

Produced Mairead Meagher
by: Dr. Siobhán Drohan



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

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

Topics list

- Why arrays?
- Primitive Arrays
- Syntax

Why arrays?

- We look at different pieces of code to explain the concept.
- In each case:
 - we read in 10 numbers from the keyboard
 - add them
 - print the result.

Adding 10 numbers

```
int n;  
int sum = 0;  
for (int i = 0; i<10; i++) {  
    n = Integer.parseInt  
        (JOptionPane.showInputDialog(  
            "Please enter a number ", "3"));  
    sum += n;  
}  
println("The sum of the values you typed in is : " + sum);
```

- Reads in 10 numbers from the keyboard
- Adds the numbers
- Prints out the result
- Does not remember the numbers

Rule – Never lose data

- Should always try to store that data for later use (in a more real-life system you would almost always need to use the input data again).
- The previous code has not done this.
- We could try another way ...

Remembering the Numbers

```
int n0,n1, n2, n3, n4, n5, n6, n7, n8, n9;  
int sum = 0;  
  
n0 = Integer.parseInt (JOptionPane.showInputDialog("Please enter a number ", "3"));  
sum += n0;  
  
n1 = Integer.parseInt (JOptionPane.showInputDialog("Please enter a number ", "3"));  
sum += n1;  
  
//rest of code for n2 to n8  
  
n9= Integer.parseInt(JOptionPane.showInputDialog("Please enter a number ", "3"));  
sum += n9;  
  
println("The sum of the values you typed in is : " + sum);
```

Remembering the Numbers

```
int n0,n1, n2, n3, n4, n5, n6, n7, n8, n9;  
int sum = 0;  
  
n0 = Integer.parseInt(JOptionPane.showInputDialog("Please enter a number ", "3"));  
sum += n0;  
  
n1 = Integer.parseInt(JOptionPane.showInputDialog("Please enter a number ", "3"));  
sum += n1;  
  
//rest of code for n2 to n8  
  
n9= Integer.parseInt(JOptionPane.showInputDialog("Please enter a number ", "3"));  
sum += n9;  
  
println("The sum of the values you typed in is : " + sum);
```

- This works in the sense that we have retained the input data.
- But:
 - We cannot use loops.
 - If we had to read in 1,000 numbers, this would require extensive code.
- We need another, new data structure...
 - **enter arrays...**

Arrays (fixed-size collections)

- Arrays are a way to collect associated values.
- Programming languages usually offer a special **fixed-size collection** type: an *array*.
- Java arrays can store objects or primitive-type values.
- Arrays use a special syntax.

Single box

If you think of a variable (field, local variable) as a box in memory:

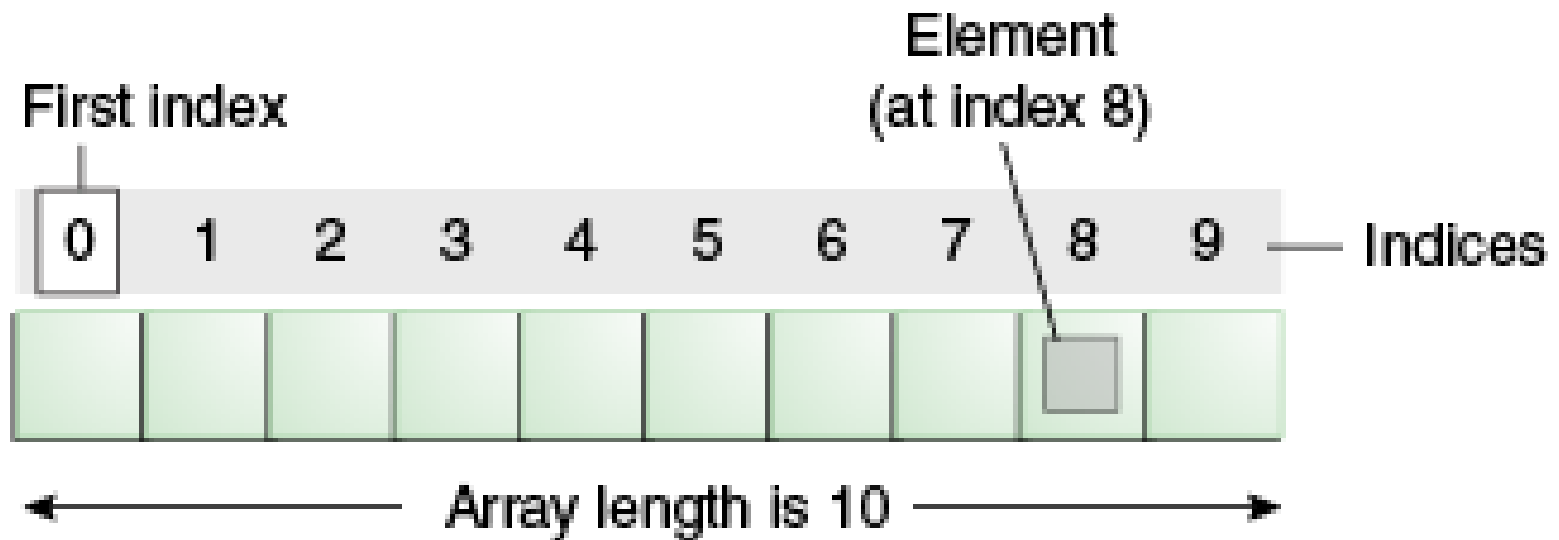
int x;



A box called 'x' in which we can put one integer

- We can:
 - change the value stored completely,
 - add one to it,
 - subtract one from it etc.
- However this box can hold only one value. Imagine a bigger box made up of sub-divisions or sections. Such a box is called an array and would look like:

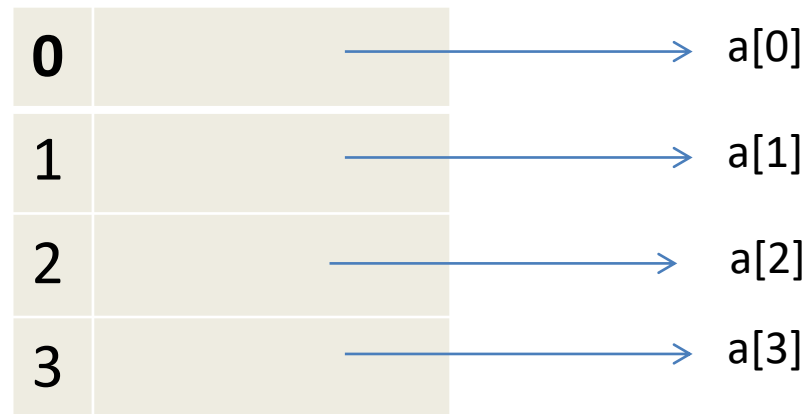
Structure of a primitive array



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.**

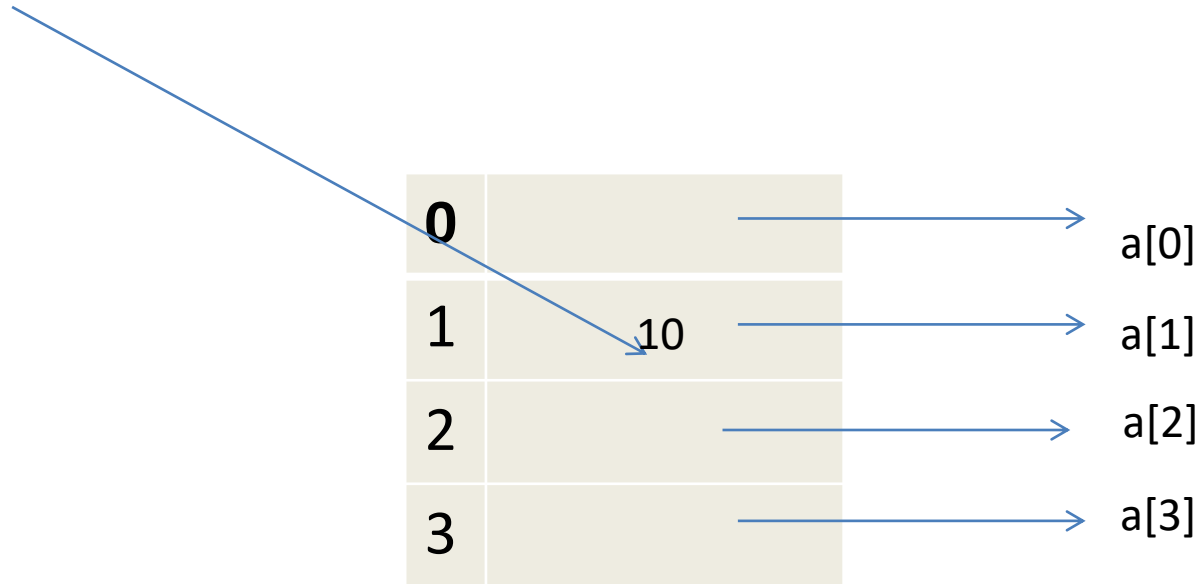
```
int[] a;  
a = new int[4];
```



Accessing elements of an array

- You can access any element separately, e.g.

`a[1] = 10;`



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];
```

Declaring primitive arrays

Can either have:

```
int[] a;
```

```
//some code here
```

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

Or

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

Returning to our method that reads
in and sums 10 numbers (typed in
from the keyboard)...

and converting it to use primitive arrays...

Using arrays to remember numbers

```
import javax.swing.*;

int a[] = new int[10];
int sum = 0;

for (int i = 0; i < 10 ; i ++ ) {
    a[i] = Integer.parseInt(JOptionPane.showInputDialog(
        "Please enter a number ", "3"));
}

for (int i = 0; i < 10 ; i ++ ) { // now we sum the values
    sum += a[i];
}

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

- Using arrays
- **Separate loop to add up the numbers**

Using arrays with any size

```
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.

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];
```

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++)
- For now, though, we assume that all elements of the array are populated and therefore ready to be processed.

Questions?





Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

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