



Unit 2: Programming

Topic 5: Develop Code

Arrays in Java

Lecture Contents



- Review of Arrays - Definition
 - Arrays in Java
 - Declaring and Initializing an Array
 - Reading an Array
 - Writing to an Array
 - Length of an Array
 - Summary
 - Practice Assignment – Linear Search in Java
- 

Array – Definition

- **Array** – a collection of elements in a **contiguous** block of memory
 - The size of the array is **fixed**
 - *fixed* = determined when it is created, and cannot be changed
 - Elements are of the same type (all numbers, or all strings, etc.)
 - Each element accessed by its **index**

This should make more sense with examples...

Array – Definition

- An array of numbers:

myNumbers

1	3	5	7	9
---	---	---	---	---

- An array of strings:

myStrings

apple
banana
peach

Declaring an Array

numbers

1	3	5	7	9
---	---	---	---	---

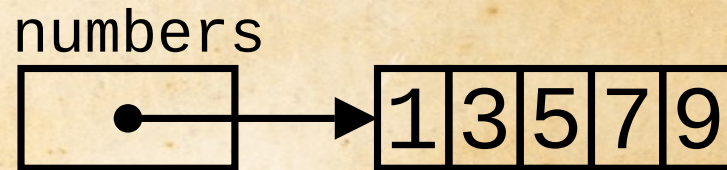
- Defining an array
 - Pearson pseudocode:
SET numbers TO [1, 3, 5, 7, 9]
 - You were also shown how to define this array in Java, Python, and C#:
 - Java: **int[] numbers = { 1, 3, 5, 7, 9 };**
 - C#: **int[] numbers = { 1, 3, 5, 7, 9 };**
 - Python: **numbers = [1, 3, 5, 7, 9]**

Declaring an Array in Java

- In Java, an array variable is actually a *reference* to where the array is stored in memory:

```
int[] numbers = { 1, 3, 5, 7, 9 };
```

- The statement above creates a structure in computer memory that can be visualized with this diagram:



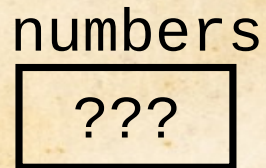
- The variable `numbers` stores the location of the array.

Declaring an Array in Java

- As with other variables, we can ***declare*** an array without ***initialization***.

```
int[] numbers;
```

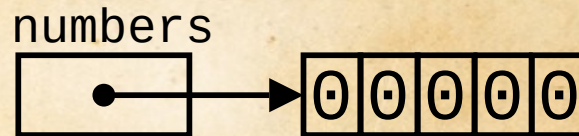
- The statement above creates a structure in computer memory that can be visualized with this diagram:



- The variable `numbers` stores an unknown ***reference*** value, and it is an error to try to read the value before it is initialized.

Declaring an Array in Java

- We can ***initialize*** an array to point to a space in memory ready to store values using the **new** keyword.
- We must tell the compiler how many ***elements*** we will need to store:
`int[] numbers = new int[5];`
- The statement above creates a structure in computer memory that can be visualized with this diagram:

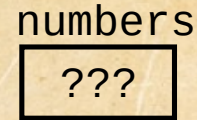


- In Java, the elements of an array will be initialized to zero (0) if it is created using the **new** keyword (boolean will be initialized to **false**).

Declaring an Array in Java

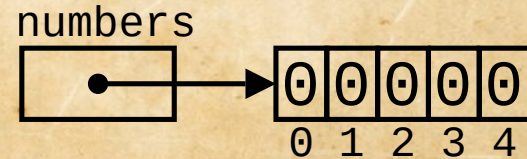
- So after we declare an array variable...

```
int[] numbers;
```



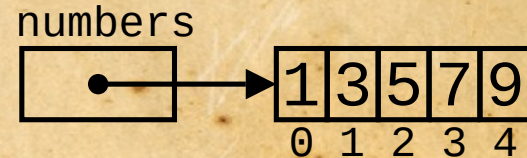
- We can either initialize it to all zeros with the keyword `new`.

```
numbers = new int[5];
```



- Or we can initialize it with values by surrounding comma separated values with curly braces:

```
numbers = { 1, 3, 5, 7, 9 };
```



Note: each number below each array element represents an **index**, which is not stored in memory.

Reading an Array

numbers

1	3	5	7	9
0	1	2	3	4

- Reading an array
 - Pearson pseudocode:
SET numbers TO [1, 3, 5, 7, 9]
SET value TO numbers[2]
SEND value TO DISPLAY
 - Array indexes (indices) are zero-based in most programming languages
 - not zero-based in R, COBOL, MATLAB, and a few others
 - So the output of the above pseudocode will be: **5**

Reading an Array

numbers

1	3	5	7	9
0	1	2	3	4

- Reading an array
 - Pearson pseudocode:
 - **SET value TO numbers[2]**
 - You have seen how to read an element of this array in Java, Python, and C#:
 - Java: **value = numbers[2];**
 - C#: **value = numbers[2];**
 - Python: **value = numbers[2]**

Writing to an Array

numbers

1	3	5	7	9
0	1	2	3	4

- Writing to an array
 - Pearson pseudocode:
 - **SET numbers[2] TO 6**

numbers

1	3	6	7	9
---	---	---	---	---

- And you've seen how to write a value this array in Java, Python, and C#:
 - Java: **numbers[2] = 6;**
 - C#: numbers[2] = 6;
 - Python: numbers[2] = 6

Length of an Array

numbers

1	3	5	7	9
0	1	2	3	4

- Getting the length of an array
 - Pearson pseudocode:
 - **SET length TO LENGTH(numbers)**
 - How to get the length of an array in Java, Python, and C#:
 - Java: **length = numbers.length;**
 - C#: **length = numbers.Length;**
 - Python: **length = len(numbers)**

Arrays of Different Types

- We can create arrays of any type, including ***primitive types***

```
int[] numbers = new int[5];
```

```
double[] reals = { 0.1, 3.4, 2.3, 9.1, 0.0 };
```

```
boolean[] values = new boolean[3];
```

```
char[] answers = { 'a', 'b', 'c', 's' };
```

- And ***objects***, such as strings

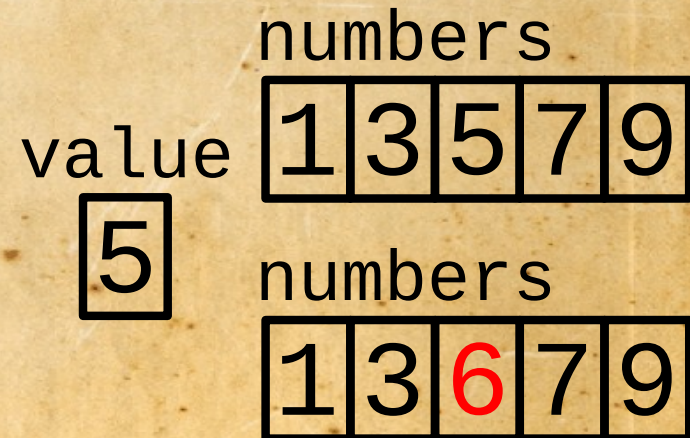
```
String[] names = { "Andy", "Brian", "Chris" };
```

Arrays – Summary

- **Array** – a collection of elements in a **contiguous** block of memory
 - The size of the array is **fixed**
 - *fixed* = determined when it is created, and cannot be changed
 - Elements are of the same type (all numbers, or all strings, etc.)
 - Each element accessed by its **index**

Pseudocode:

```
SET numbers TO [ 1, 3, 5, 7, 9 ]  
SET value TO numbers[ 2 ]  
• SET numbers[ 2 ] TO 6  
SEND value TO DISPLAY
```

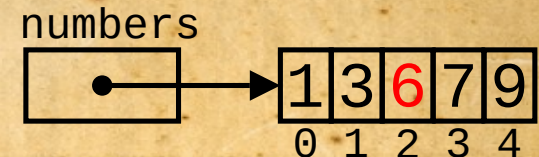
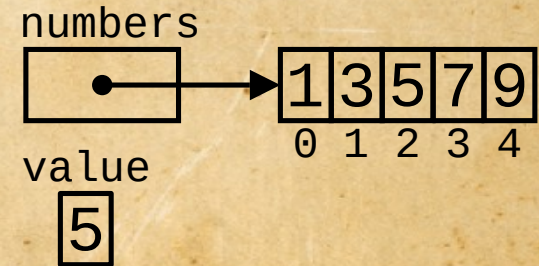


Arrays – Summary

- **Array** – a collection of elements in a **contiguous** block of memory
 - The size of the array is **fixed**
 - *fixed* = determined when it is created, and cannot be changed
 - Elements are of the same type (all numbers, or all strings, etc.)
 - Each element accessed by its **index**

Java:

```
int[] numbers = { 1, 3, 5, 7, 9 };  
int value = numbers[ 2 ];  
numbers[ 2 ] = 6;  
System.out.print(value);
```





Unit 2: Programming

Topic 5: Develop Code

Arrays in Java