

# Introduction : Arrays

- An array is a list of similar things
- An array has a fixed:
  - name
  - type
  - length
- These must be declared when the array is created.
- Arrays sizes cannot be changed during the execution of the code

myArray = 

<b>3</b>	<b>6</b>	<b>3</b>	<b>1</b>	<b>6</b>	<b>3</b>	<b>4</b>	<b>1</b>
0	1	2	3	4	5	6	7

myArray has room for 8 elements

- the elements are accessed by their index
- in Java, array indices start at 0

# Declaring Arrays

```
int myArray[];
```

declares *myArray* to be an array of integers

```
myArray = new int[8];
```

sets up 8 integer-sized spaces in memory, labelled *myArray[0]* to *myArray[7]*

```
int myArray[] = new int[8];
```

combines the two statements in one line

# Scope: Assigning Values

- refer to the array elements by index to store values in them.

```
myArray[0] = 3;
```

```
myArray[1] = 6;
```

```
myArray[2] = 3; ...
```

- can create and initialise in one step:

```
int myArray[] = {3, 6, 3, 1, 6, 3, 4, 1};
```

# Iterating Through Arrays

- *for* loops are useful when dealing with arrays:

```
for (int i = 0; i < myArray.length;
    i++) {
    myArray[i] = getsomevalue();
}
```

# Arrays of Objects

- So far we have looked at an array of primitive types.
  - integers
  - could also use doubles, floats, characters...
- Often want to have an array of objects
  - Students, Books, Loans .....
- Need to follow 3 steps.

# Declaring the Array

1. Declare the array

```
private Student studentList[];
```

- this declares studentList

2. Create the array

```
studentList = new Student[10];
```

- this sets up 10 spaces in memory that can hold references to Student objects

3. Create Student objects and add them to the array:

```
studentList[0] = new Student("Cathy",  
"Computing");
```