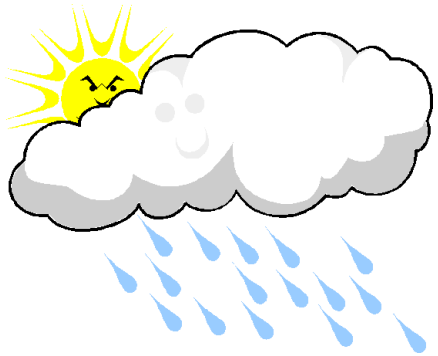
Arrays

Can we solve this problem?

- Consider the following program (input underlined):

```
How many days' temperatures? 7
Day 1's high temp: 45
Day 2's high temp: 44
Day 3's high temp: 39
Day 4's high temp: 48
Day 5's high temp: 37
Day 6's high temp: 46
Day 7's high temp: 53
Average temp = 44.6
4 days were above average.
```



Why the problem is hard

- We need each input value twice:
  - to compute the average (a cumulative sum)
  - to count how many were above average

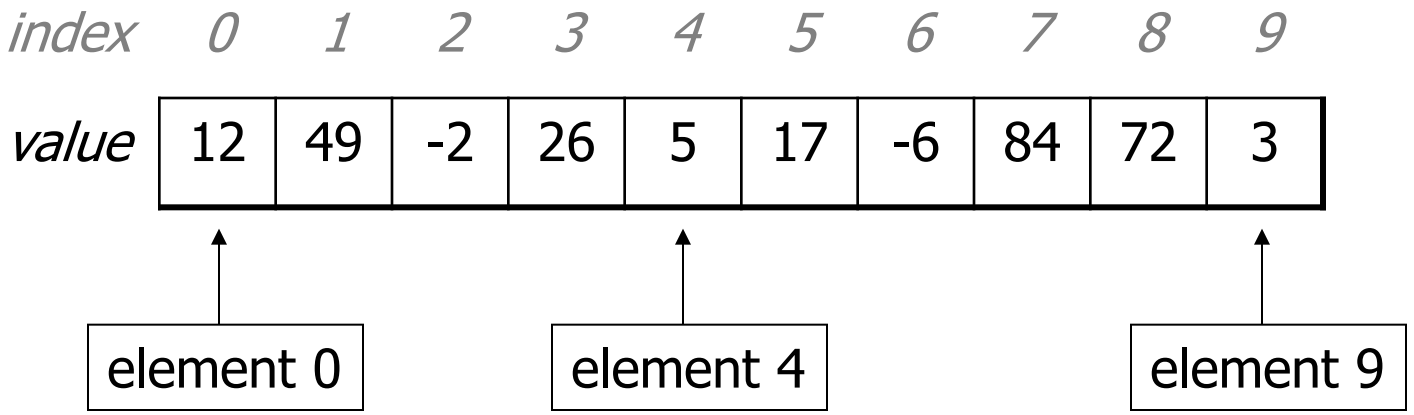
- We could read each value into a variable... but we:
  - don't know how many days are needed until the program runs
  - don't know how many variables to declare

- We need a way to declare many variables in one step.

Arrays

- A programmer commonly needs to maintain a list of items.
- **Array:** is an ordered list of items of a given data type.
- **Element**: Each item in an array is called an **element**.

-index: A 0-based integer to access an element from an array.



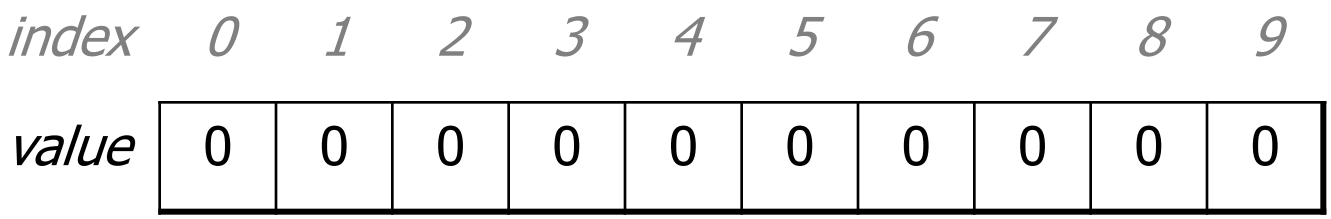
Array declaration

Format:

```
type[] name = new type[length];
```

–Example:

```
int[] numbers = new int[10];
```



Array declaration, cont.

- The length can be any integer expression.

```
int x = 2 * 3 + 1;
int[] data = new int[x % 5 + 2];
```

- Each element initially gets a "zero-equivalent" value.

Type	Default value
`int`	`0`
`double`	`0.0`
`boolean`	`false`
`String` or other object	`null` (means, "no object")

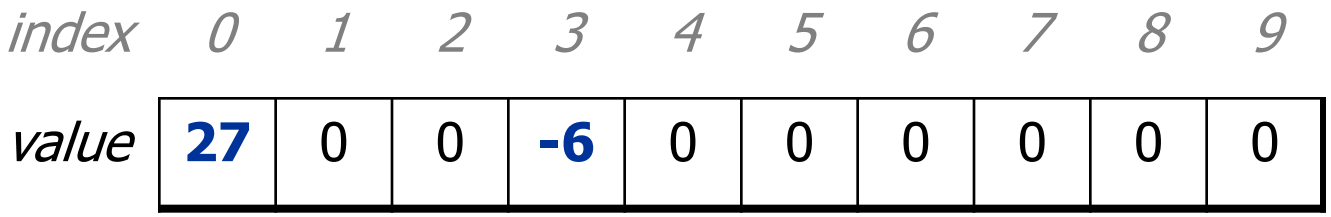
Accessing elements

```
name[index]                    // access
name[index] = value;           // modify
```

– Example:

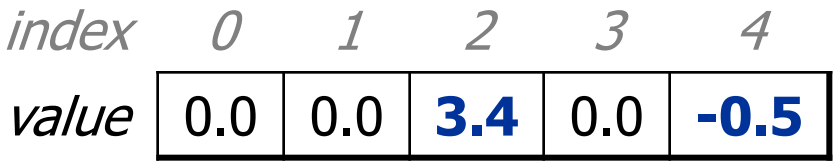
```
numbers[0] = 27;
numbers[3] = -6;

System.out.println(numbers[0]);
if (numbers[3] < 0) {
    System.out.println("Element 3 is negative.");
}
```

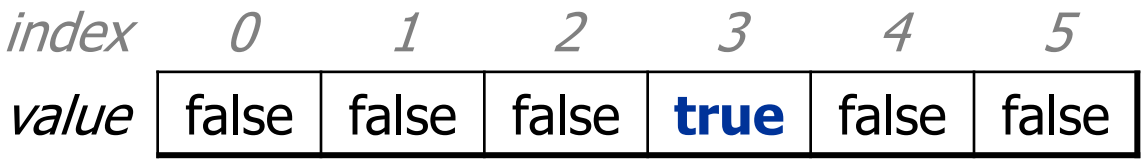


Arrays of other types

```
double[] results = new double[5];
results[2] = 3.4;
results[4] = -0.5;
```




```
boolean[] tests = new boolean[6];
tests[3] = true;
```

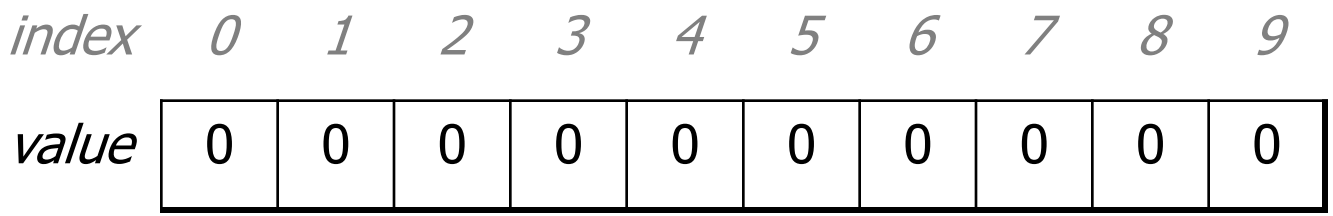


Out-of-bounds

- Legal indexes: between **0** and the **array's length - 1**.
  - Reading or writing any index outside this range will throw an `ArrayIndexOutOfBoundsException`.

- Example:

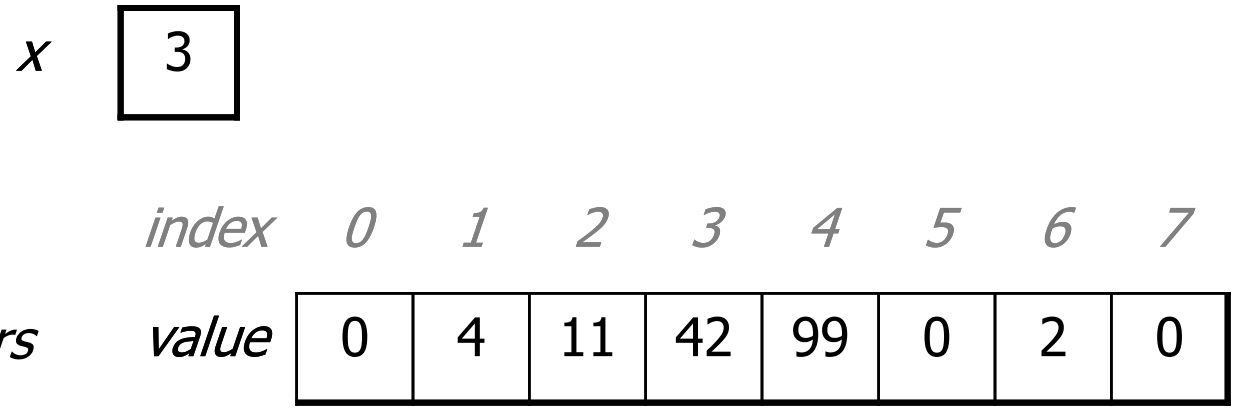
```
int[] data = new int[10];
System.out.println(data[0]);        // okay
System.out.println(data[9]);        // okay
System.out.println(data[-1]);       // exception
System.out.println(data[10]);       // exception
```



Accessing array elements

```
int[] numbers = new int[8];
numbers[1] = 3;
numbers[4] = 99;
numbers[6] = 2;

int x = numbers[1];
numbers[x] = 42;
numbers[numbers[6]] = 11; // use numbers[6] as index
```



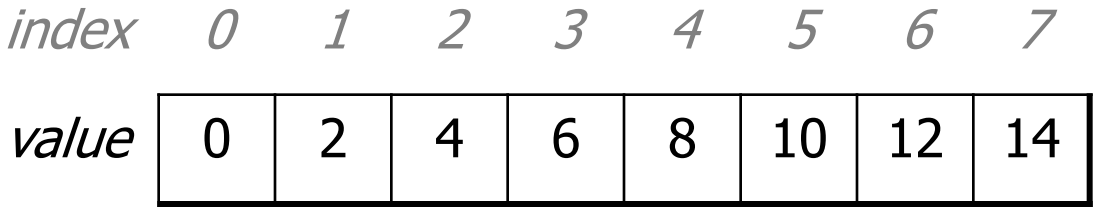
Arrays and `for` loops

- It is common to use `for` loops to access array elements.

```
for (int i = 0; i < 8; i++) {
    System.out.print(numbers[i] + " ");
}
System.out.println();  // output: 0 4 11 0 44 0 0 2
```

- Sometimes we assign each element a value in a loop.

```
for (int i = 0; i < 8; i++) {
    numbers[i] = 2 * i;
}
```



The `length` field

- An array's `length` field stores its number of elements.

  name`.length`

```
for (int i = 0; i < numbers.length; i++) {
    System.out.print(numbers[i] + " ");
}
// output: 0 2 4 6 8 10 12 14
```

  - It does not use parentheses like a String's `.length()`.

- What expressions refer to:
  - The last element of any array?
  - The middle element?

Weather question

- Use an array to solve the weather problem:

```
How many days' temperatures? 7
Day 1's high temp: 45
Day 2's high temp: 44
Day 3's high temp: 39
Day 4's high temp: 48
Day 5's high temp: 37
Day 6's high temp: 46
Day 7's high temp: 53
Average temp = 44.6
4 days were above average.
```

Weather answer

```
// Reads temperatures from the user, computes average and # days above average.
import java.util.*;

public class Weather {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        System.out.print("How many days' temperatures? ");
        int days = console.nextInt();

        int[] temps = new int[days];         // array to store days' temperatures
        int sum = 0;

        for (int i = 0; i < days; i++) {     // read/store each day's temperature
            System.out.print("Day " + (i + 1) + "'s high temp: ");
            temps[i] = console.nextInt();
            sum += temps[i];
        }
        double average = (double) sum / days;

        int count = 0;                       // see if each day is above average
        for (int i = 0; i < days; i++) {
            if (temps[i] > average) {
                count++;
            }
        }

        // report results
        System.out.printf("Average temp = %.1f\n", average);
        System.out.println(count + " days above average");
    }
}
```

Quick array initialization

type`[]` **name** `=` `{`**value**`,` **value**`,` ... **value**`};`

– Example:

```
int[] numbers = {12, 49, -2, 26, 5, 17, -6};
```

index	*0*	*1*	*2*	*3*	*4*	*5*	*6*
value	12	49	-2	26	5	17	-6

– Useful when you know what the array's elements will be

– The compiler figures out the size by counting the values

Common for loop structure for iterating through an array.

```
// Iterating through myArray

for (i = 0; i < myArray.length; ++i) {

// Loop body accessing myArray[i]

}
```

Printing array elements.

CHALLENGEACTIVITY 6.2.2: Printing array elements.

Write three statements to print the first three elements of array runTimes. Follow each statement with a newline. Ex: If runTimes = {800, 775, 790, 805, 808}, print:

```
800
775
790
```

```
import java.util.Scanner;

public class PrintRunTimes {
   public static void main (String [] args) {
      Scanner scnr = new Scanner(System.in);
      final int NUM_ELEMENTS = 5;
      int [] runTimes = new int[NUM_ELEMENTS];
      int i;

      for (i = 0; i < runTimes.length; ++i) {
         runTimes[i] = scnr.nextInt();
      }

       System.out.println(runTimes[0]);
       System.out.println(runTimes[1]);
       System.out.println(runTimes[2]);

   }
}
```

Printing array elements with a for loop.

CHALLENGEACTIVITY. 6.2.3

Write a for loop to print all elements in courseGrades, following each element with a space (including the last). Print forwards, then backwards. End each loop with a newline. Ex: If courseGrades = {7, 9, 11, 10}, print:

```
7 9 11 10
10 11 9 7
```

```
import java.util.Scanner;

public class CourseGradePrinter {
   public static void main (String [] args) {
      Scanner scnr = new Scanner(System.in);
      final int NUM_VALS = 4;
      int [] courseGrades = new int[NUM_VALS];
      int i;

      for (i = 0; i < courseGrades.length; ++i) {
         courseGrades[i] = scnr.nextInt();
      }

       for (i = 0; i < courseGrades.length; ++i) {
   System.out.print(courseGrades[i] + " ");
}
System.out.println("");

for (i = courseGrades.length - 1; i >= 0; --i) {
   System.out.print(courseGrades[i] + " ");
}
System.out.println("");

   }
}
```

"Array mystery" problem

- **traversal**: An examination of each element of an array.
- What element values are stored in the following array?

```
int[] a = {1, 7, 5, 6, 4, 14, 11};
for (int i = 0; i < a.length - 1; i++) {
    if (a[i] > a[i + 1]) {
        a[i + 1] = a[i + 1] * 2;
    }
}
```

index	*0*	*1*	*2*	*3*	*4*	*5*	*6*
value	1	7	10	12	8	14	22

Limitations of arrays

- You cannot resize an existing array:

```
int[] a = new int[4];
a.length = 10;              // error
```

- You cannot compare arrays with == or `equals`:

```
int[] a1 = {42, -7, 1, 15};
int[] a2 = {42, -7, 1, 15};
if (a1 == a2) {  ... }              // false!
if (a1.equals(a2)) {  ... }         // false!
```

- An array does not know how to print itself:

```
int[] a1 = {42, -7, 1, 15};
System.out.println(a1);             // [I@98f8c4]
```

Print the sum and average of an array's elements.

```
port java.util.Scanner;

public class ArraySum {
   public static void main(String[] args) {
      Scanner scnr = new Scanner(System.in);
      final int NUM_ELEMENTS = 8;            // Number of elements
      int[] userVals = new int[NUM_ELEMENTS]; // User numbers
      int i;                                 // Loop index
      int sumVal;                            // For computing sum

      // Prompt user to populate array
      System.out.println("Enter " + NUM_ELEMENTS + " integer values...");

      for (i = 0; i < userVals.length; ++i) {
         userVals[i] = scnr.nextInt();
         System.out.println("Value: " + userVals[i]);
      }

      // Determine sum
      sumVal = 0;
      for (i = 0; i < userVals.length; ++i) {
         sumVal = sumVal + userVals[i];
      }
      System.out.println("Sum: " + sumVal);
   }
}
```

```
3 5 234 346 234 73 26 -1
```

Run

```
Enter 8 integer values...
Value: 3
Value: 5
Value: 234
Value: 346
Value: 234
Value: 73
Value: 26
Value: -1
Sum: 920
```

```
import java.util.Scanner;

public class ArraySum {
   public static void main(String[] args) {
      Scanner scnr = new Scanner(System.in);
      final int NUM_ELEMENTS = 8;            // Number of elements
      int[] userVals = new int[NUM_ELEMENTS]; // User numbers
      int i;                                 // Loop index
      int sumVal;                            // For computing sum

      // Prompt user to populate array
      System.out.println("Enter " + NUM_ELEMENTS + " integer values...");

      for (i = 0; i < userVals.length; ++i) {
         userVals[i] = scnr.nextInt();
         System.out.println("Value: " + userVals[i]);
      }

      // Determine sum
      sumVal = 0;
      for (i = 0; i < userVals.length; ++i) {
         sumVal = sumVal + userVals[i];
      }
      System.out.println("Sum: " + sumVal);
      System.out.println("Average: " + sumVal/NUM_ELEMENTS);
   }
}
```

```
3 5 234 346 234 73 26 -1
```

Run

```
Enter 8 integer values...
Value: 3
Value: 5
Value: 234
Value: 346
Value: 234
Value: 73
Value: 26
Value: -1
Sum: 920
```

Populating an array with a for loop.

- Write a for loop to populate array userGuesses with NUM_GUESSES integers. Read integers using Scanner.

Ex: If NUM_GUESSES is 3 and user enters 9 5 2, then userGuesses is {9, 5, 2}.

Solution:

```
import java.util.Scanner;

public class StoreGuesses {
   public static void main (String [] args) {
      Scanner scnr = new Scanner(System.in);
      final int NUM_GUESSES = 3;
      int[] userGuesses = new int[NUM_GUESSES];
      int i;

        for (i = 0; i < userGuesses.length; ++i) {
       userGuesses[i] = scnr.nextInt();
        }

      for (i = 0; i < userGuesses.length; ++i){
         System.out.print(userGuesses[i] + " ");
      }
   }
}
```

Array iteration: Sum of excess.

- Array testGrades contains NUM_VALS test scores. Write a for loop that sets sumExtra to the total extra credit received. Full credit is 100, so anything over 100 is extra credit. Ex: If testGrades = {101, 83, 107, 90}, then sumExtra = 8, because 1 + 0 + 7 + 0 is 8.

```
import java.util.Scanner;

public class SumOfExcess {
   public static void main (String [] args) {
      Scanner scnr = new Scanner(System.in);
      final int NUM_VALS = 4;
      int[] testGrades = new int[NUM_VALS];
      int i;
      int sumExtra = -9999; // Assign sumExtra with 0 before your for loop

      for (i = 0; i < testGrades.length; ++i) {
         testGrades[i] = scnr.nextInt();
      }

   sumExtra = 0;
    for (i = 0; i < testGrades.length; ++i) {
    if (testGrades[i] > 100) {
   sumExtra = sumExtra + (testGrades[i] - 100);
   }
}

      System.out.println("sumExtra: " + sumExtra);
   }
}
```