PAGE 1: Creating arrays

```
// Topic 1: Creating an Array
//          Native Data Types

// an array of 10 integers
int[] scores = new int[10];


// an array of 99 booleans
boolean[] modes = new boolean[99];


// Note: default value of 0 or false
```

```
// Problems 1: Create arrays
//             Native Data Types

// an array of 5 decimal-point numbers



// an array of 800 text characters



// Note: default value of 0.0, but not '0'
```

```
// Topic 2: Creating an Array
//          Objects

// an array of 9 strings
// initial strings are blank
String[] lines = new String[9];
int i = 0;
for (i = 0; i < lines.length; i++) {
  lines[i] = "";
}




// an array of 100 Color objects
// initial color is black
Color[] crayons = new Color[100];
int i = 0;
for (i = 0; i < crayons.length; i++) {
  crayons[i] = new Color(0, 0, 0);
}


// Note: default value of null when
// objects are not initialized
```

```
// Problems 2: Create arrays
//             Objects

// an array of 1000 strings named classes
// initial string values of ELN 101
```

```
// an array of 5 Point objects named pts
// initial x value of width
// initial y value of height
```

```
// Topic 3: Creating an Array
//          using method calls

// an array of strings
// one entry per line in file
// there must be a file named data.txt

String[] lines = loadStrings("data.txt");
```

```
// Topic 3: Creating an Array
//          using method calls

// get all lines from file transcript.txt
// store in array named trscr
```

```
// Topic 4: Creating a new array from old

int[] nums = new int[10];

// oops, now I need 20 new numbers

nums = new int[20];
```

```
// Topic 4: Creating a new array from old
// Create an array of 5 booleans


// now reset it for 50 booleans
```

PAGE 2: Initializing Arrays

```
// Topic 5: Initializing an array
//          Native Data Types

// nums is an array of integers
// let's set all array values to -1
int i = 0;
for (i = 0; i < nums.length; i++) {
  nums[i] = -1;
}
```

```
// Problems 5: Initialize arrays
//             Native Data Types

// gpas is an array of doubles
// let's set all array values to 3.14




// modes is an array of Booleans
// let's set first and last value to true
// let's set the rest to false
```

```
// Topic 6: Initializing an array
//          Native Data Types

// nums is an array of integers
// let's set nums to be 1, 2, 3, etc.
// so b[0] is 1, b[1] is 2, etc.
int i = 0;
for (i = 0; i < nums.length; i++) {
  nums[i] = i + 1;
}
```

```
// Problems 6: Initialize arrays
//             Native Data Types

// decs is an array of doubles
// let's set decs to 0, 3.1, 6.2, 9.3, etc.




// decs is an array of doubles
// let's set decs to 1, 0.5, 0.25, etc.
```

PAGE 3: Analysis Using Arrays

```
// Topic 7: Sum array numbers

// nums is an array of integers
// let's find the sum of the numbers
int sum = 0;
int i = 0;
for (i = 0; i < nums.length; i++) {
  sum = sum + nums[i];
}
```

```
// Problem 7: Count array numbers

// nums is an array of integers
// determine how many negative numbers
// are in the array
```

```
// Topic 8: Find the biggest value

// nums is an array of integers
// let's find the biggest number
int biggest = nums[0];
int i = 0;
for (i = 1; i < nums.length; i++) {
  if (nums[i] > biggest) {
    biggest = nums[i];
  }
}

System.out.println(biggest);
```

```
// Problem 8: Find shortest string

// lines is an array of strings
// find the length of the shortest string
```

PAGE 4: Manipulating Arrays

```
// Topic 9: Everyone gets extra credit

// grades is an array of integers
// everyone gets 10 extra points
int i = 0;
for (i = 0; i < grades.length; i++) {
  grades[i] = grades[i] + 10;
}
```

```
// Problem 9: Everyone gets bad news

// gpas is an array of doubles
// everyone's gpa gets cut in half
```

```
// Topic 10: Mystery code

// nums is an array of integers
// what does this code do?
int i = 0;
for (i = 1; i < nums.length; i++) {
  nums[i] = nums[i] + nums[i-1];
}
```

```
// Problem 10: Peer pressure

// dbls is an array of doubles
// LOOP 1
// make a new array of doubles as follows:
//   name: ppds
//   size: same as dbls
//   values: average of neighbors
// example: dbls is 1.0 2.0 6.0 4.0 8.0
//          ppds is 1.5 3.0 4.0 6.0 6.0
// LOOP 2
// now copy each ppds[i] into dbls[i]
```

More Array Practice on Codingbat

http://codingbat.com/java/Array-1 (no loops)

http://codingbat.com/java/Array-2 (loops)

challenges: centeredAverage, has22, has12, shiftLeft, pre4, zeroFront, evenOdd