Declaring Arrays

- Arrays as we're looking at them occupy a fixed amount of space in memory.
- The programmer specifies the type of each element and the number of elements required by each array so that the compiler may reserve the appropriate amount of space in memory for the array.

Declaring Arrays

- To declare an array of 12 `int` elements:
  ```java
  int[] c = new int[12];
  ```

- To declare two `float` arrays of differing sizes:
  ```java
  float[] fred = new float[668];
  barney = new float[15];
  ```
  or
  ```java
  float[] fred, barney;
  fred = new float[668];
  barney = new float[15];
  ```
Examples: Basic Declaration / Access

1. public class Array1 {  
2.     public static void main ( String args[] ) {  
3.         int[] n;  
4.         n = new int[10];  
5.         System.out.printf ("%5s%8s\n",  
6.             "Index", "Value");  
7.         for ( int x = 0; x < n.length; x++ )  
8.             System.out.printf ("%5d%8d\n",  
9.                 x, n[x]);  
10.     }  
11. }  

Examples: Basic Declaration / Access

1. public class Array2 {  
2.     public static void main ( String args[] ) {  
3.         int[] n;  
4.         n = new int[4];  
5.         System.out.printf ("%5s%8s\n", "Index","Value");  
6.         System.out.printf ("%5d%8d\n", 0, n[0]);  
7.         System.out.printf ("%5d%8d\n", 1, n[1]);  
8.         System.out.printf ("%5d%8d\n", 2, n[2]);  
9.         System.out.println("   3   " + n[3]);  
10.     }  
11. }  

Examples: Modification of elements

1. public class Array3 {  
2.     public static void main ( String args[] ) {  
3.         int[] n = new int[10];  
4.         for ( int x = 0; x < n.length; x++ )  
5.             n[x] = x * 10;  
6.         System.out.printf ("%5s%8s\n", "Index","Value");  
7.         for ( int x = 0; x < n.length; x++ )  
8.             System.out.printf ("%5d%8d\n", x, n[x]);  
9.     }  
10. }
Examples: Initializer List
1. public class Array4 {
2. public static void main ( String args[] ) {
3.     int[] n = {32, 27, 64, 18, 95, 14, 90, 70, 60, 37};
4.     System.out.printf ("%5s%8s\n", "Index","Value");
5.     for ( int x = 0; x < n.length; x++ )
6.         System.out.printf ("%5d%8d\n", x, n[x]);
7. }
8. }
9. }

Examples: Bad Initializer List
1. public class Array5 {
2. public static void main ( String args[] ) {
3.     int[] n = {32, 27, 64, 18, 95, 14, 90, 70, 60, 37};
4.     System.out.printf ("%5s%8s\n", "Index","Value");
5.     for ( int x = 0; x < n.length; x++ )
6.         System.out.printf ("%5d%8d\n", x, n[x]);
7. }
8. }
9. }
10. // this code will not compile, as line 4 is wrong

Examples: Bad Initializer List
1. public class Array6 {
2. public static void main ( String args[] ) {
3.     int[] n = new int[10];
4.     n[] = {32, 27, 64, 18, 95, 14, 90, 70, 60, 37};
5.     System.out.printf ("%5s%8s\n", "Index","Value");
6.     for ( int x = 0; x < n.length; x++ )
7.         System.out.printf ("%5d%8d\n", x, n[x]);
8. }
9. }
10. // this code also won't compile due to line 4
Examples: Using final variables

```java
public class Array7 {
    public static void main ( String args[] ) {
        final int ARRAY_SIZE = 10;
        int[] s = new int[ARRAY_SIZE];
        for ( int j = 0; j < s.length; j++ )
            s[j] = 2 + 2 * j;
        System.out.printf ("%5s%8s
", "Index","Value");
        for ( int x = 0; x < s.length; x++ )
            System.out.printf ("%5d%8d
", x, s[x]);
    }
}
```

Examples: Using final variables

```java
public class Array8 {
    public static void main ( String args[] ) {
        final int ARRAY_SIZE = 1000;
        int[] s = new int[ARRAY_SIZE];
        for ( int j = 0; j < s.length; j++ )
            s[j] = 2 + 2 * j;
        System.out.printf ("%5s%8s
", "Index","Value");
        for ( int x = 0; x < s.length; x++ )
            System.out.printf ("%5d%8d
", x, s[x]);
    }
}
```

Examples: Summing an Array

```java
public class Array9 {
    public static void main ( String args[] ) {
        int[] n = {1, 3, 5, 5, 7, 2, 99, 16, 45, 67};
        int total = 0;
        for ( int i = 0; i < n.length ; i++ )
            total += n[i];
        System.out.println ("Total of values is "+ total);
    }
}
```
Examples: Array of Counters

```java
import java.util.Scanner;

public class ArrayOfCounters {
    public static void main ( String[] args ) {
        final int ARR_SIZE = 11, HOWMANY = 5;
        int[] freq = new int[ARR_SIZE];
        Scanner in = new Scanner ( System.in );
        System.out.println("Enter values between 1 and 10.");
        for ( int x = 1; x <= HOWMANY; x ++ ) {
            System.out.print("Enter #" + x + ": ");
            int value = in.nextInt();
            freq[value] ++ ;
        }
        System.out.printf( "%5s%10s
", "Value", "Frequency" );
        for ( int x = 1; x < freq.length; x ++ )
            System.out.printf( "%5s%10s
", x, freq[x] );
    }
}
```

Enhanced for statement

- There is a shorter, read-only version of the `for` loop that insures that you process every element of an array.
- This version of the `for` loop only allows you to use or examine each element, from the 1st element (element 0) to the last element (`array.length - 1`).

An enhanced for loop example

```java
public class EnhancedFor {
    public static void main ( String[] args ) {
        int[] nums = { 5, 10, 15, 20, 25, 30, 35, 40 };
        for ( int single: nums)
            System.out.println("Look, it's: " + single);
    }
}
```