

Arrays

Arrays

An *array* is simply a collection of values of the same type.

- Just like a string, a linear array is indexed from 0 to its length - 1.
- Values in an array must be of the same type.
- An array cannot grow, i.e., its length is fixed once defined.

Arrays

An array is declared like a variable in the following way:

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

This is actually an abbreviation of

```
type[] name;
```

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

Arrays

Example. To declare an array of 10 integers named `numbers`, put

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

This is an abbreviation of

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

Arrays

Example. To declare an array of 1,000 strings named `words`, put

```
String[] words = new String[1000];
```

This is an abbreviation of

```
String[] words;  
words = new String[1000];
```

Arrays

Example. A **for** loop to print all strings in the array `words`:

```
for (int i = 0; i < words.length; i++)  
    System.out.println(words[i]);
```

Searching

Example. A **for** loop to see if there is a specific value, say, `v`, in the array `numbers`:

```
for (int i = 0; i < numbers.length; i++)  
    if (numbers[i] == v) {  
        System.out.println(v + " exists at index " + i  
            + ".");  
        break;  
    }
```

Searching

Example. A **for** loop to count how many copies of a specific value, say, v , are in the array `numbers`:

```
int counter = 0;
for (int i = 0; i < numbers.length; i++)
    if (numbers[i] == v)
        counter++;
}
System.out.println("There are " + counter
    + " copies of " + v + ".");
```

Searching

Example. A **for** loop to find the *maximum* integer in the array `numbers`:

```
int maxSoFar = MININT;
for (int i = 0; i < numbers.length; i++)
    if (numbers[i] > maxSoFar)
        maxSoFar = numbers[i];
System.out.println("Max: " + maxSoFar);
```

Searching

Example. A **for** loop to find both the *maximum* and *minimum* integers in the array `numbers`:

```
int maxSoFar = MININT;
int minSoFar = MAXINT;
for (int i = 0; i < numbers.length; i++) {
    if (numbers[i] > maxSoFar) maxSoFar = numbers[i];
    if (numbers[i] < minSoFar) minSoFar = numbers[i];
}
System.out.println("Max: " + maxSoFar);
System.out.println("Min: " + minSoFar);
```

Multi-dimensional arrays

Example. To declare a 5 x 10 array of integers named `table`, put

```
int[][] table = new int[5][10];
```

Multi-dimensional arrays

Example. Nested **for** loops to print all integers in the array `table`:

```
for (int i = 0; i < 5; i++) {  
    for (int j = 0; j < 10; j++)  
        System.out.print(table[i][j] + " ");  
    System.out.println();  
}
```

Multi-dimensional arrays

Example. Nested **for** loops to print the sum of all numbers in the array `table`:

```
int sum = 0;
for (int i = 0; i < 5; i++)
    for (int j = 0; j < 10; j++)
        sum += table[i][j];
System.out.println("Sum: " + sum);
```

Multi-dimensional arrays

Example. Nested **for** loops to print the maximum number in the array `table`:

```
int maxSoFar = MININT;  
for (int i = 0; i < 5; i++)  
    for (int j = 0; j < 10; j++)  
        if (table[i][j] > maxSoFar)  
            maxSoFar = table[i][j];  
System.out.println("Max: " + maxSoFar);
```