

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 100 characters named `letters`, put

```
char[] letters = new char[100];
```

This is an abbreviation of

```
char[] letters;  
letters = new char[100];
```

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 print all *negative* integers in the array `numbers`:

```
System.out.print("Negative integers are: ");  
for (int i = 0; i < numbers.length; i++)  
    if (numbers[i] < 0)  
        System.out.print(numbers[i] + " ");
```

Searching

Example. A **for** loop to see if there is a *negative* integer in the array `numbers`:

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

Searching

Example. A **for** loop to see if there is an *even* integer in the array `numbers`:

```
for (int i = 0; i < numbers.length; i++)  
    if (numbers[i] % 2 == 0) {  
        System.out.println("An even integer, namely, "  
            + numbers[i] + ", exists at index " + i + ".");  
        break;  
    }
```

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 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 the *maximum even integer* in the array `numbers`:

```
int maxSoFar = MININT;

for (int i = 0; i < numbers.length; i++)
    if ((numbers[i] % 2 == 0) && (numbers[i] > maxSoFar))
        maxSoFar = numbers[i];
System.out.println("Max even: " + maxSoFar);
```

What if there are no even numbers?

Searching

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

```
int maxSoFar = MININT;
boolean found = false;
for (int i = 0; i < numbers.length; i++)
    if ((numbers[i] % 2 == 0) && (numbers[i] > maxSoFar)) {
        maxSoFar = numbers[i];
        if (!found) found = true;
    }
if (found) System.out.println("Max even: " + maxSoFar);
else System.out.println("No even integer exists.");
```