

An array is simply a sequence of values of the same type. The values stored in an array are called the elements of the array. The length of an array is determined by the number of elements in the array. For example, an array with 10 elements has a length of 10. The elements in an array are referenced by their index, which ranges from 0 to the length of the array-1. This means, in order to reference the object in the second position in an array "array", the user would call "array[1]". An array can contain values of any type, not just primitives, but any objects as well. In order to construct an array, the programmer must supply an element type, the length of the array, and a variable name, used to reference the array in memory. The length of an array, once instantiated is fixed, and additional elements cannot be added. The length of an array "array" can be called using the expression "array.length". The declaration of an array called "number", which stores 10 integers looks like this:

```
int[] number = new int[10]
```

When created, an array originally contains all 0s, falses, or nulls, depending on the element type. If a user knows which values an array should contain prior to construction, an array can also be generated using a list of values as such:

```
int[] number = {1,2,3,4,5,6,7,8,9,10}
```

If this is not the case, and a programmer wishes to either give or change the values of the elements in an array, this can be done quite easily. The array stores a list of references to the spaces in memory which contain the elements of the array. Elements can be assigned values by referencing them by their index:

```
number[2] = 3
```

When parallel arrays (multiple arrays that contain elements which are related to one another) exist in a program, it is almost always better to use an array of objects instead. Arrays can be extended to multiple dimensions as well. A two-dimensional array is often used to store values of a matrix or table. A two-dimensional array is basically an array of arrays. When creating a two-dimensional array, programmers must specify the length of both the rows and columns in the array. Elements in the array are referenced based on first their row index, then their column index in separate brackets. For example, this is how to construct a 2-D array with 3 rows and 5 columns.

```
int[][] table = new int[3][5]
```

In addition, it is possible to reference an entire row in a 2-D array by providing only the row value. In this way, it is possible to create a 2-D array with rows of variable length. Such arrays might be useful for problems dealing with triangular matrixes. Arrays are not limited to two dimensions.

An array is a very useful form of storing values, for one, because it makes sorting and searching incredibly easy. Referencing similar objects by number, rather than name is also neater than cramming 100 different variable names into a program. Arrays are incredibly useful when used in combination with for loops, because the variables are already referenced by index, which is exactly what a for loop uses.

ArrayLists, unlike standard arrays, are a class that must be imported from the Java Framework. Also unlike regular arrays, ArrayLists can be increased in length as needed. This prevents bugs caused by fixed length arrays being overfilled with input data. ArrayLists can store any type or element, including null. Thus, it is possible to create multi-dimensional ArrayLists in much the same way you create multi-dimensional arrays. Also, ArrayLists must be modified using the ArrayList methods. If the ArrayList's internal array is modified directly, it will throw an error. The ArrayList class has a variety of methods for searching, comparing, and editing its elements, such as indexOf, contains, add, and set. Other than the outlined differences, ArrayLists serve the same functions as standard arrays, and have similar properties.