Arrays

OS x OD

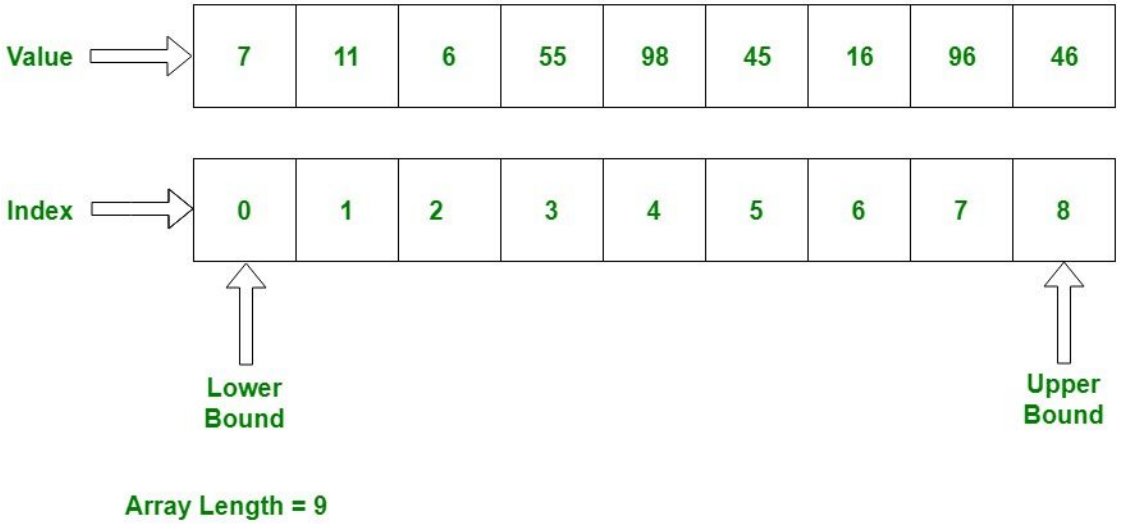
What are arrays

An array is a collection of individual data values with two distinguishing characteristics:

- An array is ordered. You must be able to count off the values: here is the first, here is the second, and so on
- An array is homogeneous. Every value in the array must have the same type.

An array is an ordered list of values

- The entire array has a single name
- Each value has a numeric index
- An array of size N is indexed from zero to N-1



Arrays

A particular value in an array is referenced using the array name followed by the index in brackets

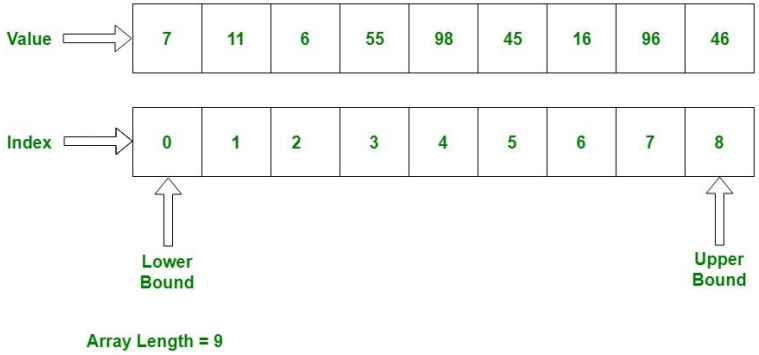
For example, the expression

scores[2]

refers to the value 6 (the 3rd value in the array)

That expression represents a place to store a single integer and can be used wherever an integer variable can be used

Array name = scores



Using arrays

For example, an array element can be assigned a value, printed, or used in a calculation:

```
scores[2] = 89;

scores[first] = scores[first] + 2;

mean = (scores[0] + scores[1])/2;

System.out.println ("Top = " + scores[5]);
```

Arrays

The values held in an array are called array elements

An array stores multiple values of the same type (the element type)

The element type can be a primitive type or an object reference

Therefore, we can create an array of integers, or an array of characters, or an array of String objects, etc.

In Java, the array itself is an object

Therefore the name of the array is a object reference variable, and the array itself must be instantiated

Declaring Arrays

The scores array could be declared as follows:

int[] scores = new int[10];

The type of the variable scores is int[] (an array of integers)

Note that the type of the array does not specify its size, but each object of that type has a specific size

The reference variable scores is set to a new array object that can hold 10 integers

Some examples of array declarations:

Java

```
float[] prices = new float[500];

boolean[] flags;

flags = new boolean[20];

char[] codes = new char[1750];
```

Python

```
# empty array
arr = []
# init with values (can contain mixed types)
arr = [1, "eels"]
# get item by index (can be negative to access end of array)
arr = [1, 2, 3, 4, 5, 6]
```

C++

```
// Array declaration by specifying size
int arr1[10];

// With recent C/C++ versions, we can also
// declare an array of user specified size
int n = 10;
int arr2[n];
```

Bounds Checking

Once an array is created, it has a fixed size

An index used in an array reference must specify a valid element

That is, the index value must be in bounds (0 to N-1)

The Java interpreter throws an ArrayIndexOutOfBoundsException if an array index is out of bounds

This is called automatic bounds checking

Bounds Checking

For example, if the array codes can hold 100 values, it can be indexed using only the numbers 0 to 99

If count has the value 100, then the following reference will cause an exception to be thrown:

System.out.println (codes[count]);

It’s common to introduce off-by-one errors when using arrays

Length of Arrays

Each array object has a public constant called length that stores the size of the array

It is referenced using the array name:

scores.length

Note that length holds the number of elements, not the largest index

Alternate Array Syntax

The brackets of the array type can be associated with the element type or with the name of the array

Therefore the following declarations are equivalent:

```
float[] prices;

float prices[];
```

The first format generally is more readable

Initializer List

An initializer list can be used to instantiate and initialize an array in one step

The values are delimited by braces and separated by commas

Examples:

```
int[] units = {147, 323, 89, 933, 540,
               269, 97, 114, 298, 476};

char[] letterGrades = {'A', 'B', 'C', 'D', 'F'};
```

Initializer List

Note that when an initializer list is used:

the new operator is not used

no size value is specified

The size of the array is determined by the number of items in the initializer list

An initializer list can only be used only in the array declaration

What is units[5]?

```
int[] units = {147, 323, 89, 933, 540,
               269, 97, 114, 298, 476};
```

Change the first element in the array to equal 291

```
int[] units = {147, 323, 89, 933, 540,

                269, 97, 114, 298, 476};
```

Take 10 integer inputs from user and store them in an array and print them on screen

```
Scanner s = new Scanner(System.in);

int[] z = new int[10];

for(int i = 0;i<z.length;i++){
  System.out.println("Print the value of z["+i+"]");
  z[i] = s.nextInt();
}

for(int i = 0;i<z.length;i++){
  System.out.println("The value of z["+i+"] is "+z[i]);
}
```

Write a Java program to sum values of an array.

```
public class Exercise2 {

public static void main(String[] args) {

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

int sum = 0;


for (int i : my_array)

    sum += i;

System.out.println("The sum is " + sum);

}

}
```

Hackerrank

https://www.hackerrank.com/challenges/java-1d-array-introduction/problem

Practice Problems

Java

https://www.codesdope.com/practice/java-array/

Python

https://www.w3resource.com/python-exercises/array/

C++

https://www.w3resource.com/cpp-exercises/array/index.php