Arrays

An array in Java represents a number of variables which occupy contiguous spaces in the memory. Each element in the array is distinguished by its index.

Objectives

Upon completion of this lecture, we will be able to:

- Discuss array representation in Java

Arrays

All elements in an array must be of the same data type. For example, you cannot have one element with int data type and another belonging to the boolean data type in the same array. An array is a collection of elements of the same type that are referenced by a common name. Each element of an array can be referred to by an array name and a subscript or index. To create and use an array in Java, you need to first declare the array and then initialize it.

One-dimensional arrays

A one-dimensional array is a list of like-typed variables. The general form of a one dimensional array declaration is:

type var-name [ ];

Here, type declares the base type of the array and determines the data type of each element that makes the array. Hence the base type determines the type of data the array will hold. For example, **int numbers[ ];** declares an array named numbers with the type “array of int”:

This declaration says that the numbers are an array variable but no array actually exists because the value of numbers is set to null, which represents an array with no value. To link numbers with an actual, physical array of integers, you must allocate one using **new** (which is a special operator to allocate memory) and assign it to numbers. Here new is used to allocate memory for arrays. The general form of new as it applies to one-dimensional arrays is as follows:

array-var = new type[size];

Here, type specifies the type of data being allocated, size specifies the number of elements in the array, and array-var is the array variable that is linked to the array. That is, to use **new** to allocate an array, you must specify the type and number of elements to allocate. The elements in the array allocated by **new** will automatically be initialized to zero.

For example, numbers = new int[10]; allocates a 10-element array of integers and links them to numbers.

Note that in Java all arrays are dynamically allocated.

For example, numbers[1] = 23; assigns the value 23 to the second element of numbers.

Multidimensional arrays

In Java, multidimensional arrays are actually arrays of arrays. To declare a multidimensional array variable, each additional index using another set of square brackets is specified.

For example, the int mat[ ][ ] = new int[3][7];

declares a two-dimensional array variable called mat. This allocates a 3 by

7 array and assigns it to mat. Internally this matrix is implemented as an array of arrays of int.

Summary

Here are the key takeaways:

- An array is a data structure where elements are sequentially stored.
- Each element has an index.
- Using index, any element can be accessed randomly.
- An array has a read only property called length. This represents the number of elements in the array.