**Array is:**

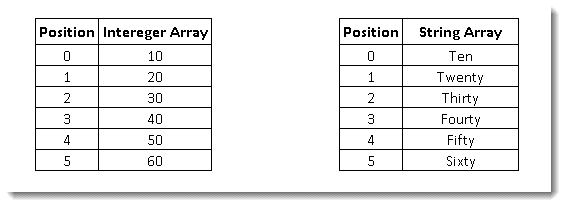
a container object and a built-in data structure

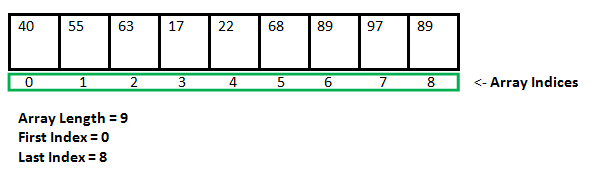
a special data structure used to store a collection of data

stores a fixed-size of elements

holds value / variable of the same data type (except an Object array)

elements of an array occupy a contiguous memory location

index-based, the first element of the array is stored at the 0 index



**Declaring an Array:**

**<data type> [] <any name of array> = new <data type> [size];** // preferred

Example: **int** [] anyArray = **new** **int** [10];

or

**<data type> <any name of array>[];** // works but not preferred

Example: **double myList [];**

**dataType[] arrayRefVar = {value0, value1, ..., value*n*};**

String[] group = **new** String[] { "a", "b", "c" };

Object [] [] anyObject = **new** Object[3][3];    // Object class array

Array length = size of array

**Accessing the element inside the array:**

String[] group = new String[] { "a", "b", "c" };

   System.out.println(group[0]);

**for each example:**

public static void main(String[] args) {

double[] myList = {1.9, 2.9, 3.4, 3.5};

**3 ways to print elements of an array:**

(i) for loop,

(ii) Arrays.toString() method and

(iii) forEach (enhanced for loop)

**Print all the array elements using Enhanced  for loop ( or advanced or forEach loop):**

for (double element: myList) {

System.out.println(element);

}

  }

**Copy all the elements of an array to another:**

**public static void main(String[] args) {**

**Integer[] intArray = new Integer[] { 1, 2, 3, 4, 5 };**

**Object[] objectArray = new Object[5];**

**System.arraycopy(intArray, 0, objectArray, 0, 5);**

**}**

**}**

**Multi-dimensional array (an example of a 2-dimensional array):**

public static void main(String[] args) {

        int[][] arr = { { 1, 2 }, { 3, 4 } };

          for (int i = 0; i < 2; i++) {

            for (int j = 0; j < 2; j++) {

                System.out.print(arr[i][j] + " ");

            }

            System.out.println();

        }

    }

**Printing length of a 2-dimensional array:**

System.out.println(numbers[0].length);   // for the first array: index is **0**

System.out.println(numbers[1].length);   // for the first array: index is **1**

**To get a specified value / element from a 2-dimensional array:**

package test.java;

public class TwoDArray {

public static void main(String[] args) {

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

// to find the value of 6

// 6 belongs to the first array at the 1 index position and

// the position of 6 is at 2 index position

System.out.println(numbers[1][2]);

}

}

**Sorting an array elements:**

package classthree;

import java.util.Arrays;

public class SortingAnArrayElements {

public static void main(String[] args) {

int[] nums = { 10, 2, 3, 1, 59, 61, 17, 8, 91, 0 };

Arrays.sort(nums);        // sort() method

System.out.println(Arrays.toString(nums));

     }

}

**Printing the max and min value of an array after sorting:**

package arraydemo;

import java.util.Arrays;

import java.util.Collections;

public class ArrayDemo {

public static void main(String[] args) {

int[] numbers = { 2, 34, 5, 65, 76, 889, 643, 32, 782, 3, 0 };

Arrays.sort(numbers); // sorting in an ascending manner

System.out.println(numbers[(numbers.length) - 1]); // from last index

System.out.println(numbers[0]); // from the first index

System.out.println(Arrays.toString(numbers)); // printing

}

}

**A List:12/8/19**

is another type of container object that hold similar types of data

list is a child interface of collection

it contains index-based methods to insert and delete elements

can be printed directly

**Declaring a list:**

List<DataType> nameOfTheList = new List<DataType>();

        List<String> students = new ArrayList<String>();

**Creating a list of integers:**

List<Integer> numbers = new ArrayList<Integer>();

**Add**ing elements to a list of integers:

numbers.**add**(11);

numbers.add(22);

numbers.add(33);

numbers.add(44);

**adding elements to a list of integers using index position:**

numbers.add(**0**, **4**);          // add **4** at index position **0**

students.add(**1**, “**Sapin**”);   // add **Sapin** at index position **1**

**adding a null element:**

students.add(**null**); // add **null**

**adding a null elements at an index position:** // add **null** at index position **10**

students.add(**10**, null);

**remove an element from a List:**

students.**remove**(“Nepal”);  // remove **Nepal**

students.remove(**0**);   // remove element from index position **0**

**Clear** or remove all elements from a list:

students.**clear**();

**size of a list:**

int size = numbers.**size**();

**Getting an element / item from a list:**

numbers.**get**(0);                             // from a specific index

students.indexOf("Suman");             // first index of “Suman”

source.lastIndexOf("Kumar"); // last index of “Suman”

**printing all the elements of a list:**

                      System.out.println(numbers);

**printing all the elements of a list using for each loop(advanced or enhanced of loop)**

for(String s : numbers){

System.out.println(s);

}

}

**printing all the elements of a list using Iterator:**

                      Iterator<String> it = numbers.iterator();

        while(it.hasNext()){

            Object a = it.next();

            System.out.println(a);

        }

**sorting the elements of a list:**

Collections.sort(numbers);

**insert all elements from one list into another:**

listDest.addAll(listSource);

**check if “Sapin” is in the list:**

students.contains("Sapin");

**A list of objects that holds various types of data:**

public class ListPractice {

public static void main(String[] args) {

List<Object> etc = new ArrayList<Object>();

etc.add("Thapa");

etc.add(12);

etc.add(2.14);

etc.add(12);

etc.add(124335345L);

etc.add('c');

Iterator<Object> myIt = etc.iterator();

while(myIt.hasNext()) {

Object a = myIt.next();

System.out.println(a);

}

}

}

**Output:**

Thapa

12

2.14

12

124335345

c