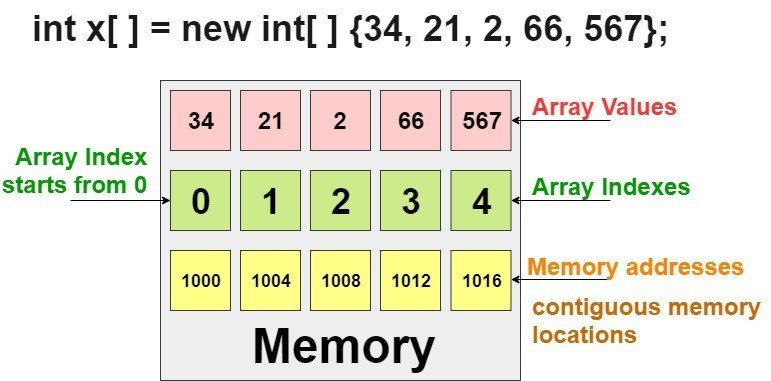
**Arrays**



1. Normally, array is a collection of similar type of elements that have contiguous memory location.  
  
2. Java array is an object the contains elements of similar data type. It is a data structure where we store similar elements. We can store only fixed set of elements in a java array.  
  
3. Array in java is index based, first element of the array is stored at 0 index.  
  
**Advantage of Java Array:**  
  
**1. Code Optimization:**It makes the code optimized, we can retrieve or sort the data easily.  
  
**2. Random access:**We can get any data located at any index position.  
  
**Disadvantage of Java Array:**  
**Size Limit:**We can store only fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in java.

**There are two types of array:**  
1. Single Dimensional Array  
2. Multidimensional Array

**Single Dimensional Array**  
  
**Syntax to Declare an Array in java:**  
  
dataType[] arr; (or)    
dataType []arr; (or)    
dataType arr[];    
  
Instantiation of an Array in java:  
  
arrayRefVar=new datatype[size];    
  
class Testarray{    
  
public static void main(String args[ ]){    
int a[ ]=new int[5];//declaration and instantiation    
a[0]=10;//initialization    
a[1]=20;    
a[2]=70;    
a[3]=40;    
a[4]=50;    
//printing array    
for(int i=0;i<a.length;i++)//length is the property of array    
System.out.println(a[i]);    
}  
  }    
**Output:**  
10  
20  
70  
40  
50  
  
**We can declare, instantiate and initialize the java array together by:**  
  
int a[]={33,3,4,5};//declaration, instantiation and initialization    
  
Let's see the simple example to print this array.

class Testarray1{    
public static void main(String args[]){    
    
int a[]={33,3,4,5};//declaration, instantiation and initialization    
    
//printing array    
for(int i=0;i<a.length;i++)//length is the property of array    
System.out.println(a[i]);    
    
}

## }

## Output: 33 3 4 5

## Multidimensional Array in Java

In such case, data is stored in row and column based index (also known as matrix form).

**Syntax to Declare Multidimensional Array in Java**

dataType[][] arrayRefVar; (or)

dataType [][]arrayRefVar; (or)

dataType arrayRefVar[][]; (or)

dataType []arrayRefVar[];

**Example to instantiate Multidimensional Array in Java**

**int**[][] arr=**new** **int**[3][3];//3 row and 3 column

**Example to initialize Multidimensional Array in Java**

arr[0][0]=1;

arr[0][1]=2;

arr[0][2]=3;

arr[1][0]=4;

arr[1][1]=5;

arr[1][2]=6;

arr[2][0]=7;

arr[2][1]=8;

arr[2][2]=9;

**Example of Multidimensional Java Array**

Let's see the simple example to declare, instantiate, initialize and print the 2Dimensional array.

//Java Program to illustrate the use of multidimensional array

**class** Testarray3{

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

//declaring and initializing 2D array

**int** arr[][]={{1,2,3},{2,4,5},{4,4,5}};

//printing 2D array

**for**(**int** i=0;i<3;i++){

**for**(**int** j=0;j<3;j++){

System.out.print(arr[i][j]+" ");

 }

 System.out.println();

}

}}

**Output:**

1 2 3

2 4 5

4 4 5

**Three Dimentional Array**

|  |
| --- |
| class GFG {      public static void main(String[] args)      {            int[][][] arr = { { { 1, 2 }, { 3, 4 } }, { { 5, 6 }, { 7, 8 } } };            for (int i = 0; i < 2; i++)              for (int j = 0; j < 2; j++)                  for (int z = 0; z < 2; z++)                      System.out.println("arr[" + i                                         + "]["                                         + j + "]["                                         + z + "] = "                                         + arr[i][j][z]);      }  } |

**Output:**

arr[0][0][0] = 1

arr[0][0][1] = 2

arr[0][1][0] = 3

arr[0][1][1] = 4

arr[1][0][0] = 5

arr[1][0][1] = 6

arr[1][1][0] = 7

arr[1][1][1] = 8

Simple example of 3d Array

**package** Evening;

**import** Morg.First;

**public** **class** Third **extends** First{

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

String[ ] department = {"Electronics", "CS", "IT"};

**int** dept, st, sc, total = 0;

**double** perc = 0;

// Take the scores of students in a 3D array.

**int**[ ][ ][ ] scores = {

{{75, 87, 69}, {90, 87, 85},{56, 67, 76}},

{{78, 67, 75}, {87, 98, 76}, {67, 56, 66}},

{{72, 63, 72}, {82, 91, 71}, {64, 56, 66}}

};

// Display the scores of students from 3D array.

**for**(dept = 0; dept < 3; dept++)

{

**for**(**int** i = 0; i < 3; i++)

{

System.***out***.println("Department " +department[i]+ ": ");

**for**(st = 0; st < 3; st++)

{

System.***out***.println("Student" +(st + 1)+ " scores: ");

**for**(sc = 0; sc < 3; sc++)

{

System.***out***.print(scores[dept][st][sc]+ " ");

total += scores[dept][st][sc];

perc = (**double**)total/3;

}

System.***out***.println("\nTotal scores: " +total); // Displaying total marks of student.

System.***out***.println("Percentage: " +perc); // Displaying percentage.

System.***out***.println();

System.***out***.println("---------------------");

System.***out***.println();

total = 0; // reset total to zero.

}

System.***out***.println();

}

**break**;

}

}

}

1. **Declare an array**

int a[];

a=new int[40];

|  |
| --- |
| String[] aArray = **new** String[5];  String[] bArray = {"a","b","c", "d", "e"};  String[] cArray = **new** String[]{"a","b","c","d","e"}; |

**1. Print an array in Java**

|  |
| --- |
| **int**[] intArray = { 1, 2, 3, 4, 5 };  String intArrayString = Arrays.toString(intArray);    *// print directly will print reference value*  System.out.println(i*ntArray*);  *// [I@7150bd4d*    System.out.println(intArrayString);  *// [1, 2, 3, 4, 5]* |

**2. Create an ArrayList from an array**

|  |
| --- |
| String[] stringArray = { "a", "b", "c", "d", "e" };  ArrayList<String> arrayList = **new** ArrayList<String>(Arrays.asList(stringArray));  System.out.println(arrayList);  *// [a, b, c, d, e]* |

**3. Check if an array contains a certain value**

|  |
| --- |
| String[] stringArray = { "abc", "bcd", "cba", "dcb", "efg" };  **boolean** b = Arrays.asList(stringArray).contains("abc");  System.out.println(b);  *// true* |

**4. Concatenate two arrays**

|  |
| --- |
| **int**[] intArray = { 1, 2, 3, 4, 5 };  **int**[] intArray2 = { 6, 7, 8, 9, 10 };  *// Apache Commons Lang library*  **int** a[] = ArrayUtils.addAll(intArray, intArray2);  String s = Arrays.toString(a);  System.out.println(s);  //1,2,3,4,5,6,7,8,9,10 |

**6. Joins the elements of the provided array into a single String**

|  |
| --- |
| *// containing the provided list of elements*  *// Apache common lang*  String j = StringUtils.join(**new** String[] { "a", "b", "c" }, ", ");  System.out.println(j);  *// a, b, c* |

**7. Covnert an ArrayList to an array**

|  |
| --- |
| String[] stringArray = { "a", "b", "c", "d", "e" };  ArrayList<String> arrayList = **new** ArrayList<String>(Arrays.asList(stringArray));  String[] stringArr = **new** String[arrayList.size()];  arrayList.toArray(stringArr);  **for** (String s : stringArr)  System.out.println(s); |

**8. Convert an array to a set**

String[] s1 = { "a", "b", "c", "d", "e",”a” };

|  |
| --- |
| Set<String> set123 = **new** HashSet<String>(Arrays.asList(s1));  System.out.println(set123);  *//[d, e, b, c, a]* |

**9. Reverse an array**

|  |
| --- |
| **int**[] intArray = { 1, 2, 3, 4, 5 };  ArrayUtils.reverse(intArray);  System.out.println(Arrays.toString(intArray));  *//[5, 4, 3, 2, 1]* |

**10. Remove element of an array**

|  |
| --- |
| **int**[] intArray = { 1, 2, 3, 4, 5 };  **int**[] removed = ArrayUtils.removeElement(intArray, 3);*//create a new array*  System.out.println(Arrays.toString(removed)); |

**COPYING AN ARRAY:**

**1. USING ARRAYCOPY() METHOD:**

The arraycopy() method belongs to the System class. It is used to copy the contents of one array to the other.

**Example:**

|  |  |
| --- | --- |
|  | public class ArrayCopying {      public static void main(String[] args) {          int Source[] = {5,6,7,8,9,10};          int Destination[] = new int[5];          System.arraycopy(Source, 1, Destination, 0, 5);          System.out.print(Arrays.toString(Destination));  }  } |
|  |  |

**OUTPUT:**

|  |
| --- |
| [6, 7, 8, 9, 10] |

**2. USING COPYOF() AND COPYOFRANGE() METHOD:**

The Arrays class has two commonly used methods for copying the content of array i.e. copyOf() and copyOfRange() method.

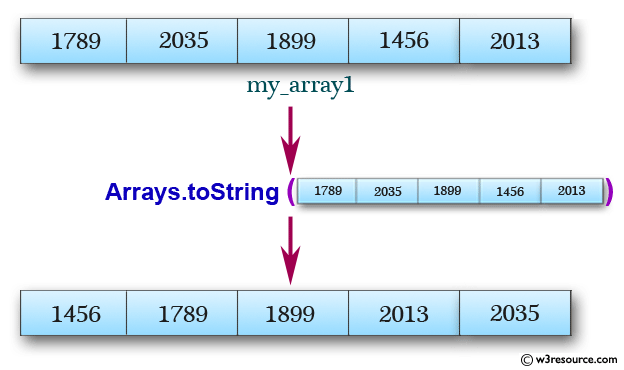
**EXAMPLE:**

|  |  |
| --- | --- |
|  | public class ArrayCopying {      public static void main(String[] args) {          int Source[] = {5,6,7,8,9,10};            int Destination[] = java.util.Arrays.**copyOf(**Source, 8);//0 is padded for index beyond source            System.out.println(Arrays.toString(Destination));            int Destination2[] = java.util.Arrays.**copyOfRange**(Source, 0, 5);            System.out.println(Arrays.toString(Destination2));        }  } |

**OUTPUT:**

|  |
| --- |
| [5, 6, 7, 8, 9, 10, 0, 0] [5, 6, 7, 8, 9] |

**Write a Java program to sort a numeric array and a string array.**



import java.util.Arrays;

public class Exercise1 {

public static void main(String[] args){

int[] my\_array1 = {

1789, 2035, 1899, 1456, 2013,

1458, 2458, 1254, 1472, 2365,

1456, 2165, 1457, 2456};

String[] my\_array2 = {

"Java",

"Python",

"PHP",

"C#",

"C Programming",

"C++"

};

System.out.println("Original numeric array : "+Arrays.toString(my\_array1));

Arrays.sort(my\_array1);

System.out.println("Sorted numeric array : "+Arrays.toString(my\_array1));

System.out.println("Original string array : "+Arrays.toString(my\_array2));

Arrays.sort(my\_array2);

}

}