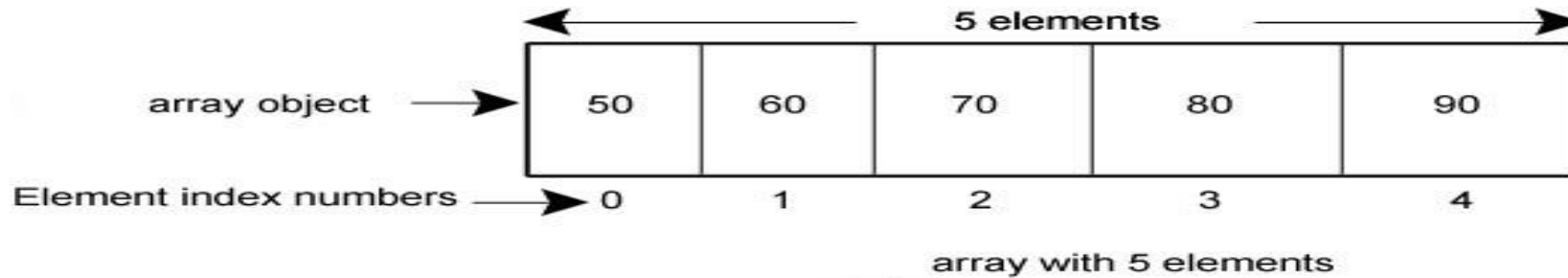# ARRAYS

1. An array is a data structure, which can store a fixed-size collection of elements of the same data type

2. Each item in array is known as element.

3. Arrays are index based and index starts from 0 to the size -1. Index are being used to represent individual elements available in the array.

4. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.

5. Two types of array
    5. Single dimensional
    6. Two dimensional

6. In java, Arrays are objects of java.util.Arrays class.

7. In the Java programming language, an array of char is not a String, and neither a String nor an array of char is terminated by the NUL character.

8. Advantage of Java Array
    5. Code Optimization: It makes the code optimized, we can retrieve or sort the data easily.
    6. Random access: We can get any data located at any index position.

9. Disadvantage of Java Array
    5. Size Limit: We can store only fixed size of elements in the array. It doesn't grow its size at runtime.

One Dimensional Array

array with 5 elements

Declaring an array

Syntax:

```
datatype[] arrayname;
```

Instantiation of an array

Syntax:

```
datatype[] arrayname=new datatype[array size];
```

Assigning values to the array

During Declaration

Syntax:

```
datatype[] arrayname={value1,value2,value3,….};
or
datatype[] arrayname=new datatype[]{value1,value2,value3,….};
```

After Declaration

Syntax:

```
arrayname[index]=value;
```

Two Dimensional Array

Elements are stored in row and column based index.

Number of elements is row * column

Declaration of array

Syntax:

```
datatype[][] arrayname;
```

Instantiation of an array

Syntax:

```
datatype[][] arrayname=new datatype[row size][column size];
```

Assigning values to the array

During Declaration

Syntax:

```
datatype[][] arrayname={{value1,value2},{value3,value4},{},{},…..};
```

or

```
datatype[][] arrayname=new datatype[][] {value1,value2},{value3,value4},{},..};
```

After Declaration

Syntax:

```
arrayname[row index][column index]=value;
```

Use of for loop

One Dimensional Array

```
int []a=new int[]{1,2,3,4,5,6};
for(int i=0;i<a.length;i++)
{
System.out.println(a[i]);
}
```

Two Dimensional Array

```
int [][]a=new
int[][]{{1,2,3},{4,5,6},{7,8,9}};
for(int i=0;i<a.length;i++)
{
for(int j=0;j<a[i].length;j++)
{
System.out.print(a[i][j]);
}
System.out.println();
}
```

Use of enhanced for loop

One Dimensional Array

```
int []arr=new int[]{1,2,3,4,5,6,7,8};
for(int i : arr)
{
System.out.println(i);
}
```

Two Dimensional Array

```
int [][]arr=new
int[][]{{1,2,3},{4,5,6},{7,8,9}};
for(int[] i : arr)
{
for(int j : i)
{
System.out.print(j);
}
System.out.println();
}
```

Arrays properties and methods

- As java arrays are the object of java.util.Arrays class arrays have the properties and methods of the Arrays class

Methods	Description
`binarySearch(datatype[] arrayname,datatype key)` `binarySearch(datatype[] array,datatype fromindex, datatype toindex, datatype key)`	Searches the specified array of datatype for the specified value using the binary search algorithm.
`copyOf(datatype[] arrayname,int newLength)`	Copies the specified array and the copy has the specified length.
`equals(datatype[] array1,datatype[] array2)`	Returns true if the two specified arrays are *equal* to one another.
`fill(datatype[] array, int fromIndex, int toIndex, datatype val)`	Assigns the specified char value to each element of the specified range of the specified array of chars.
`sort(datatype[] array)`	Sorts the specified array into ascending order.
`toString(datatype[] array)`	Returns a string representation of the contents of the specified array.