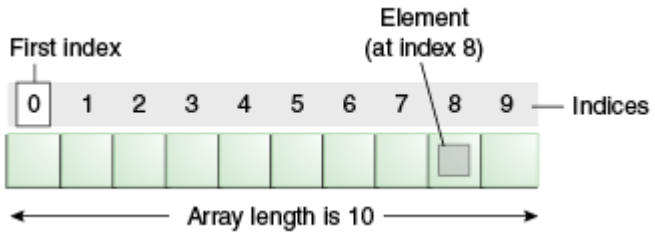
8. Java Array

Normally, array is a collection of similar type of elements that have contiguous memory location.

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.

Array in java is index based, first element of the array is stored at 0 index.



➢ Advantage of Java Array

- **Code Optimization:** It makes the code optimized, we can retrieve or sort the data easily.
- **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.

Types of Array in java

There are two types of array.

- Single Dimensional Array
- Multidimensional Array

➢ Single Dimensional Array in java

Syntax to Declare an Array in java

1. dataType[] arr; (or)
2. dataType []arr; (or)

3. dataType arr[];

Instantiation of an Array in java

1. arrayRefVar=**new** datatype[size];

Example of single dimensional java array

Let's see the simple example of java array, where we are going to declare, instantiate, initialize and traverse an array.

```
1. class Testarray{
2. public static void main(String args[]){
3.
4. int a[]=new int[5];//declaration and instantiation
5. a[0]=10;//initialization
6. a[1]=20;
7. a[2]=70;
8. a[3]=40;
9. a[4]=50;
10.
11. //printing array
12. for(int i=0;i<a.length;i++)//length is the property of array
13. System.out.println(a[i]);
14.
15. }}
```

Output:

```
10
20
70
40
50
```

➢ Declaration, Instantiation and Initialization of Java Array

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.

```
1. class Testarray1{
2. public static void main(String args[]){
```

```
3.
4. int a[]={33,3,4,5};//declaration, instantiation and initialization
5.
6. //printing array
7. for(int i=0;i<a.length;i++)//length is the property of array
8. System.out.println(a[i]);
9.
10. }}
```

Output:

```
33
3
4
5
```

Passing Array to method in java

We can pass the java array to method so that we can reuse the same logic on any array.

Let's see the simple example to get minimum number of an array using method.

```
1. class Testarray2{
2. static void min(int arr[]){
3. int min=arr[0];
4. for(int i=1;i<arr.length;i++)
5.  if(min>arr[i])
6.   min=arr[i];
7.
8. System.out.println(min);
9. }
10.
11. public static void main(String args[]){
12.
13. int a[]={33,3,4,5};
14. min(a);//passing array to method
15.
16. }}
```

Output:3

➢ 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

```
1. dataType[][] arrayRefVar; (or)
2. dataType [][]arrayRefVar; (or)
3. dataType arrayRefVar[][]; (or)
4. dataType []arrayRefVar[];
```

Example to instantiate Multidimensional Array in java

```
1. int[][] arr=new int[3][3];//3 row and 3 column
```

Example to initialize Multidimensional Array in java

```
1. arr[0][0]=1;
2. arr[0][1]=2;
3. arr[0][2]=3;
4. arr[1][0]=4;
5. arr[1][1]=5;
6. arr[1][2]=6;
7. arr[2][0]=7;
8. arr[2][1]=8;
9. arr[2][2]=9;
```

Example of Multidimensional java array

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

```
1. class Testarray3{
2. public static void main(String args[]){
3.
4. //declaring and initializing 2D array
5. int arr[][]={{1,2,3},{2,4,5},{4,4,5}};
6.
7. //printing 2D array
8. for(int i=0;i<3;i++){
9.  for(int j=0;j<3;j++){
10.   System.out.print(arr[i][j]+" ");
11. }
12. System.out.println();
13.}
14.
15.}}
```

Output:
1 2 3

```
2 4 5
4 4 5
```

➢ Addition of 2 matrices in java

Let's see a simple example that adds two matrices.

```
1. class Testarray5{
2. public static void main(String args[]){
3. //creating two matrices
4. int a[][]={{1,3,4},{3,4,5}};
5. int b[][]={{1,3,4},{3,4,5}};
6.
7. //creating another matrix to store the sum of two matrices
8. int c[][]=new int[2][3];
9.
10. //adding and printing addition of 2 matrices
11. for(int i=0;i<2;i++){
12. for(int j=0;j<3;j++){
13. c[i][j]=a[i][j]+b[i][j];
14. System.out.print(c[i][j]+" ");
15. }
16. System.out.println();//new line
17. }
18.
19. }}
```

Output:

```
2 6 8
6 8 10
```