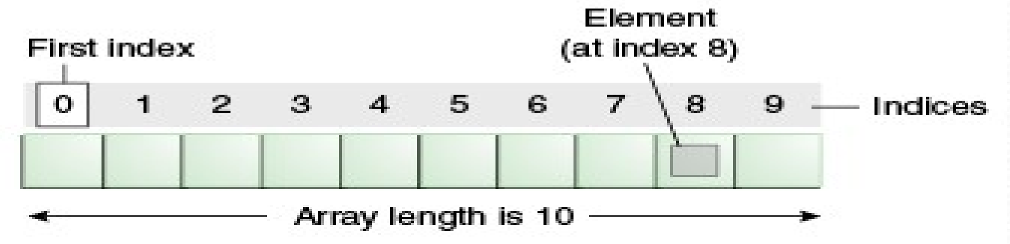
Java

Arrays

Arrays

- **Java array** is an object which contains elements of a similar data type. Additionally, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.
- Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.
- We can store primitive values or objects in an array 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**
  - dataType[] arr; (or)
  - dataType []arr; (or)
  - dataType arr[];

Single Dimensional Array

- **Instantiation of an Array in Java**
  - arrayRefVar = **new** datatype[size];

- **int** a[]=**new** int[5]; **//declaration and instantiation**

- **We can declare, instantiate and initialize the java array together by:**

  int a[]={33,3,4,5}; **//declaration, instantiation and initialization**

Single Dimensional Array

- //Java Program to illustrate how to declare, instantiate, initialize
- //and traverse the Java array.

```
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;
//traversing array
for(int i=0;i<a.length;i++)          //length is the property of array
System.out.println(a[i]);
}}
```

For-each Loop for Java Array

- We can also print the Java array using **for-each loop**. The Java for-each loop prints the array elements one by one. It holds an array element in a variable, then executes the body of the loop.

  - **for**(data_type variable:array){
  - //body of the loop
  - }

For-each Loop for Java Array

- //Java Program to print the array elements using for-each loop

```
class Testarray1{
public static void main(String args[]){
int arr[]={33,3,4,5};

//printing array using for-each loop
for(int i:arr)
System.out.println(i);
}
}
```

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[];

Multidimensional Array in Java

- **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;
```

Multidimensional Array in Java

- //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();
}
}}
```

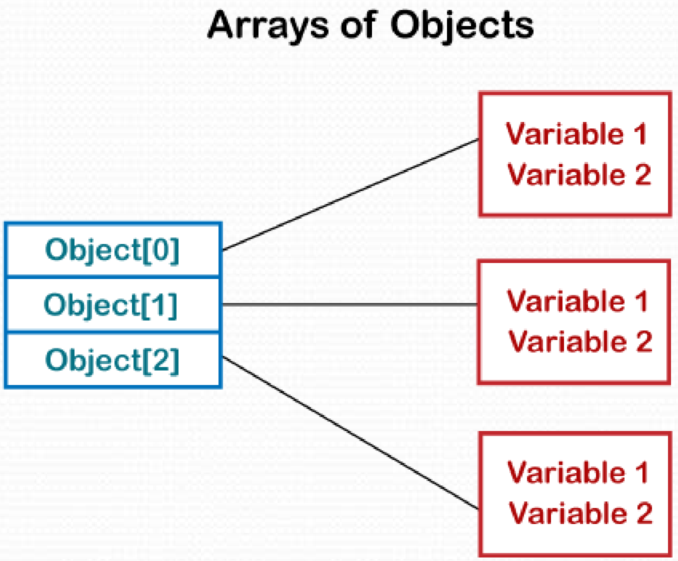
Jagged Array in Java

- If we are creating odd number of columns in a 2D array, it is known as a jagged array. In other words, it is an array of arrays with different number of columns.

```
//Java Program to illustrate the jagged array
class TestJaggedArray{
    public static void main(String[] args){
        //declaring a 2D array with odd columns
        int arr[][] = new int[3][];
        arr[0] = new int[3];
        arr[1] = new int[4];
        arr[2] = new int[2];
        //initializing a jagged array
        int count = 0;
        for (int i=0; i<arr.length; i++)
            for(int j=0; j<arr[i].length; j++)
                arr[i][j] = count++;
           //printing the data of a jagged array
        for (int i=0; i<arr.length; i++){
            for (int j=0; j<arr[i].length; j++){
                System.out.print(arr[i][j]+" ");
            }
            System.out.println();               //new line
        }     }  }
```

Array of Objects in Java

- Java allows us to store objects in an array. In Java, the class is also a user-defined data type. An array that conations **class type elements** are known as an **array of objects**. It stores the reference variable of the object.



Array of Objects in Java

- **Syntax:**

```
//declare and instantiate an array of objects
ClassName obj[]=new ClassName[array_length];
```

- ClassName obj[0] = new constructor();
- ClassName obj[1] = new constructor();
- ClassName obj[2] = new constructor();

```
public class ArrayOfObjects
{
public static void main(String args[])
{
//create an array of product object
Product[] obj = new Product[5] ;
//create & initialize actual product objects using constructor
obj[0] = new Product(23907,"Dell Laptop");
obj[1] = new Product(91240,"HP 630");
obj[2] = new Product(29823,"LG OLED TV");
obj[3] = new Product(11908,"MI Note Pro Max 9");
obj[4] = new Product(43590,"Kingston USB");
//display the product object data
System.out.println("Product Object 1:");
obj[0].display();
System.out.println("Product Object 2:");
obj[1].display();
System.out.println("Product Object 3:");
obj[2].display();
System.out.println("Product Object 4:");
obj[3].display();
System.out.println("Product Object 5:");
obj[4].display();
} }
```

```
//Product class with product Id and product name
 as attributes
class Product
{
int pro_Id;
String pro_name;
//Product class constructor
Product(int pid, String n)
{
pro_Id = pid;
pro_name = n;
}
public void display()
{
System.out.print("Product Id = "+pro_Id + "  " + "
Product Name = "+pro_name);
System.out.println();
}
}
```