UNIT-5 Arrays

TOPICS: -

5.1 One Dimensional Arrays - Declaration and Initialization, Print elements of array using enhanced for loop (for each loop)
5.2 Two Dimensional Arrays - Declaration and Initialization, Matrix Operations
5.3 Ragged Array and Multidimensional Array

ARRAYS: -

So far you have been working with variables that hold only one value. The integer variable held only one integer number and string variables just one long string of text. An array is a way to hold more than one value at a time it's like a list of items.

An array is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed.

Advantages

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

Random access: We can get any data located at an index position.

Disadvantages

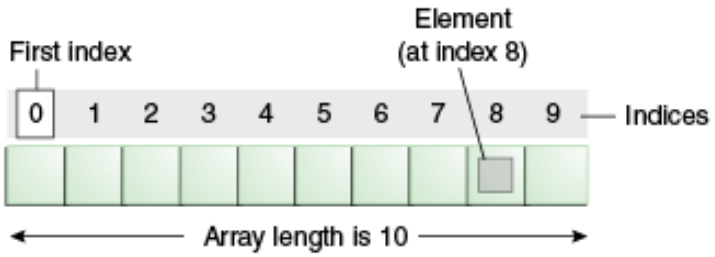
Size Limit: We can store only the fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in Java which grows automatically.

Types of Array in java: -

There are two types of array.

1. Single or One-Dimensional Array
2. Multidimensional Array

5.1 One Dimensional Arrays - Declaration and Initialization: -



Array in java is a group of like-typed variables referred to by a common name.

Creating, initializing, and accessing an Array One-Dimensional Arrays:

Declaration: -

The general form of a one-dimensional array declaration is

type var-name[];
OR
type[] var-name;

An array declaration has two components: the type and the name. type declares the element type of the array. The element type determines the data type of each element that comprises the array. Like an array of integers, we can also create an array of other primitive data types like char, float, double, etc., or user-defined data types (objects of a class). Thus, the element type for the array determines what type of data the array will hold.

Example:

```
// both are valid declarations
int Arr[];
or int[] Arr;
```

Although the first declaration establishes that Arr is an array variable, no actual array exists. It merely tells the compiler that this variable (Arr) will hold an array of the integer type. To link Arr with an actual, physical array of integers, you must allocate one using new and assign it to Arr.

Instantiating an Array in Java:-

When an array is declared, only a reference of an array is created. To create or give memory to the array, you create an array like this: The general form of new as it applies to one-dimensional arrays appears as follows:

var-name = new type [size];

Here, type specifies the type of data being allocated, size determines the number of elements in the array, and var-name is the name of the array variable that is linked to the array. To use new to allocate an array, you must specify the type and number of elements to allocate.

Example:

```
int Array[];    //declaring array
Array = new int[20];  // allocating memory to array
                OR
int[] Array = new int[20]; // combining both statements in one
```

Note:

The elements in the array allocated by new will automatically be initialized to zero (for numeric types), false (for boolean), or null (for reference types).

S. No.	Datatype	Default Value
1	boolean	false
2	int	0
3	double	0.0
4	String	null
5	User-Defined Type	null

Array Literal

In a situation where the size of the array and variables of the array are already known, array literals can be used.

```
int[] Array = new int[]{ 1,2,3,4,5,6,7,8,9,10 };  // Declaring array literal
OR
int [] Array = { 1,2,3,4,5,6,7,8,9,10};
```

- The length of this array determines the length of the created array.
- There is no need to write the new int[] part in the latest versions of Java.

Accessing Java Array Elements using for Loop :-

Each element in the array is accessed via its index. The index begins with 0 and ends at (total array size)-1. All the elements of array can be accessed using Java for Loop.

```
// accessing the elements of the specified array
for (int i = 0; i < arr.length; i++)
  System.out.println("Element at index " + i +  " : "+ arr[i]);
```

Example :- (Compile time Initialization)

```
// Java program to illustrate creating an array
// of integers, puts some values in the array,
// and prints each value to standard output.
class Create_array {
        public static void main(String[] args)
        {
                // declares an Array of integers.
                int[] arr;

                // allocating memory for 5 integers.
                arr = new int[5];

                // initialize the first elements of the array
                arr[0] = 10;

                // initialize the second elements of the array
                arr[1] = 20;

                // so on...
```

```
                arr[2] = 30;
                arr[3] = 40;
                arr[4] = 50;

                // accessing the elements of the specified array
                for (int i = 0; i < arr.length; i++)
                        System.out.println("Element at index " + i+ " : " + arr[i]);
        }
}
```

Output

```
Element at index 0 : 10
Element at index 1 : 20
Element at index 2 : 30
Element at index 3 : 40
Element at index 4 : 50
```

Example :- (Run time Initialization)

```
// Java program to illustrate creating an array
// of integers, puts some values in the array,
// and prints each value to standard output.
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
// declares an Array of integers.
int[] arr;

// allocating memory for 5 integers.
arr = new int[5];
//Enter input at runtime
for (int i = 0; i < arr.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        arr[i]=sc.nextInt();
}
// accessing the elements of the specified array
for (int i = 0; i < arr.length; i++){
  System.out.println("Element at index " + i+ " : " + arr[i]);
}
}
```

```
}
```

Output :-

```
Enter Element1: 2
Enter Element2: 6
Enter Element3:7
Enter Element4: 8
Enter Element5: 2
Element at index 0 : 2
Element at index 1 : 6
Element at index 2 : 7
Element at index 3 : 8
Element at index 4 : 2
```

Print elements of array using enhanced for loop (for each loop): -

For-each is another array traversing technique like for loop, while loop, do-while loop introduced in Java5.

Syntax:-

```
        for (int val : num)
            System.out.print(val + " ");
```

(num is an array of int type)

is equivalent to:

```
for (int i=0; i<arr.length; i++)
{
   type var = arr[i];
   statements using var;
}
```

Example: -

```
int [] arr = { 10 ,20, 12, 34, 45,50 ,60};

for (int val : arr)

        System.out.print(val + " ");
```

Output: -

10 20 12 34 45 50 60

Examples: -

```
// Java program to demonstrate default
// values of array elements

class ArrayDemo {
	public static void main(String[] args)
	{
		System.out.println("String array default values:");
		String str[] = new String[5];
		for (String s : str)
			System.out.print(s + " ");

		System.out.println(
			"\n\nInteger array default values:");
		int num[] = new int[5];
		for (int val : num)
			System.out.print(val + " ");

		System.out.println(
			"\n\nDouble array default values:");
		double dnum[] = new double[5];
		for (double val : dnum)
			System.out.print(val + " ");

		System.out.println(
			"\n\nBoolean array default values:");
		boolean bnum[] = new boolean[5];
		for (boolean val : bnum)
			System.out.print(val + " ");

		System.out.println(
			"\n\nReference Array default values:");
		ArrayDemo ademo[] = new ArrayDemo[5];
		for (ArrayDemo val : ademo)
			System.out.print(val + " ");
	}
}
```

Output: -

String array default values: null null null null null

Integer array default values: 0 0 0 0 0

Double array default values: 0.0 0.0 0.0 0.0 0.0

Boolean array default values: false false false false false

Reference Array default values: null null null null null

Programs: -

1. **Write a program to store n numbers in an array. Then find out Sum, Maximum, minimum and Average of these n numbers.**

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int sum=0,max,min;
double avg;
System.out.println("Enter N:");
int n=sc.nextInt();
int arr[]=new int[n];
for (int i = 0; i < arr.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        arr[i]=sc.nextInt();
}
max=arr[0];
min=arr[0];
for (int i = 0; i < arr.length; i++){
        sum=sum+arr[i];
        if(arr[i]>max)
                max=arr[i];
        if(arr[i]<min)
                min=arr[i];
}
avg=(double)sum/n;
System.out.println("Array Elements:");
for (int val : arr)
   System.out.print(val + " ");
System.out.println("\nSum:"+sum);
```

```
System.out.println("Minimum:"+min);
System.out.println("Maximum:"+max);
System.out.println("Average:"+avg);
}
}
```

Output: -

```
Enter N: 5
Enter Element 1: 5
Enter Element 2: 4
Enter Element 3: 2
Enter Element 4: 3
Enter Element 5: 6
Array Elements: 5 4 2 3 6
Sum:20
Minimum:2
Maximum:6
Average:4.0
```

2. Write a program to sort N integer Numbers in ascending order.

Method-1 (Using inbuilt function sort)

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter N:");
int n=sc.nextInt();
int arr[]=new int[n];
for (int i = 0; i < arr.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        arr[i]=sc.nextInt();
}
Arrays.sort(arr);
for (int val : arr)
   System.out.print(val + " ");
}
}
```

Method-2 (Without using inbuilt function)

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
```

```
int temp;
System.out.println("Enter N:");
int n=sc.nextInt();
int arr[]=new int[n];
for (int i = 0; i < arr.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        arr[i]=sc.nextInt();
}
for (int i = 0; i < arr.length; i++){
        for(int j=i+1;j<arr.length-1;j++){
                if(arr[i]>arr[j]){
                        temp=arr[i];
                        arr[i]=arr[j];
                        arr[j]=temp;
                }
        }
}
System.out.println("After Sorting Array Elements:\n");
for (int val : arr)
   System.out.print(val + " ");
}
}
```

Output: -

```
Enter N: 5
Enter Element 1: 2
Enter Element 2: 6
Enter Element 3: 4
Enter Element 4: 7
Enter Element 5: 8
After Sorting Array Elements: 2 4 6 7 8
```

3. Write a Program to accept array of N integers and find Largest odd number as well as largest even number and display them.

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int max_odd=0,max_even=0,e=0,o=0;
System.out.println("Enter N:");
int n=sc.nextInt();
int arr[]=new int[n];
```

```
for (int i = 0; i < arr.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        arr[i]=sc.nextInt();
}
for (int i = 0; i < arr.length; i++){
                if(arr[i]%2==0 && e==0){
                        max_even=arr[i];
                        e++;
                }
                if(arr[i]%2!=0 && o==0){
                        max_odd=arr[i];
                        o++;
                }
                if(arr[i]>max_even && arr[i]%2==0)
                        max_even=arr[i];
                if(arr[i]>max_odd && arr[i]%2!=0)
                        max_odd=arr[i];
}
System.out.println("Array Elements:");
for (int val : arr)
   System.out.print(val + " ");
System.out.println("\nLargest even number:"+max_even);
System.out.println("Largest odd number:"+max_odd);
}
}
```

Output: -

```
Enter N: 6
Enter Element 1: 12
Enter Element 2: 32
Enter Element 3: 45
Enter Element 4: 65
Enter Element 5: 25
Enter Element 6: 76
Array Elements:
12 32 45 65 25 76
Largest even number:76
Largest odd number:65
```

4. Write program to add two 1-D array elements in third 1-D array.

```
import java.util.*;
class Create_array  {
```

```
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter N:");
int n=sc.nextInt();
int a[]=new int[n];
int b[]=new int[n];
int c[]=new int[n];
System.out.println("Enter Element of array a:");
for (int i = 0; i < a.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        a[i]=sc.nextInt();
}
System.out.println("Enter Element of array b:");
for (int i = 0; i < b.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        b[i]=sc.nextInt();
}
for (int i = 0; i < c.length; i++){
                c[i]=a[i]+b[i];
}
System.out.println("Array c:");
for (int val : c)
   System.out.print(val + "\n");
}
}
```

Output: -

```
Enter N: 5
Enter Element of array a:
Enter Element 1: 1
Enter Element 2: 2
Enter Element 3: 3
Enter Element 4: 4
Enter Element 5: 5
Enter Element of array b:
Enter Element 1: 1
Enter Element 2: 2
Enter Element 3: 3
Enter Element 4: 4
Enter Element 5: 5
Array c:
```

```
2
4
6
8
10
```

5. Write a program read in an array of integers and print its elements in reverse order.

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter N:");
int n=sc.nextInt();
int a[]=new int[n];
System.out.println("Enter Element of array:");
for (int i = 0; i < a.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        a[i]=sc.nextInt();
}
System.out.println("Reverse array elements:");
for (int i = a.length-1; i>=0; i--){
        System.out.print(a[i]+" ");
}
}
}
```

Output: -

```
Enter N:
5
Enter Element of array:
Enter Element 1: 1
Enter Element 2: 2
Enter Element 3: 3
Enter Element 4: 4
Enter Element 5: 5
Reverse array elements:
5 4 3 2 1
```

6. Write a program to search a key from given array using for each loop.

```
import java.util.*;
class Create_array  {
```

```
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int j=0;
System.out.println("Enter N:");
int n=sc.nextInt();
int a[]=new int[n];
System.out.println("Enter Element of array:");
for (int i = 0; i < a.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        a[i]=sc.nextInt();
}
System.out.println("Enter key Element of array you want to search:");
int key=sc.nextInt();
boolean found=false;
for (int val : a){
        j++;
        if(val==key){
                found=true;
                break;
}
}
if(found==true)
        System.out.println("Key found at loacation:"+j);
else
        System.out.println("Not found");
}
}
```

Output: -

```
Enter N: 5
Enter Element of array:
Enter Element 1:1
Enter Element 2:2
Enter Element 3: 3
Enter Element 4: 4
Enter Element 5: 5
Enter key Element of array you want to search: 5
Key found at loacation:5

Enter N: 3
Enter Element of array:
```

```
Enter Element 1:1
Enter Element 2: 2
Enter Element 3: 3
Enter key Element of array you want to search: 5
Not found
```

7. Write a program to print multiple of n from given array elements using for each loop.

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter N:");
int n=sc.nextInt();
int a[]=new int[n];
System.out.println("Enter Element of array:");
for (int i = 0; i < a.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        a[i]=sc.nextInt();
}
System.out.println("Enter N which multiple you want to print:");
int N=sc.nextInt();
for (int val : a){
        if(val%N==0)
        System.out.println(val);
}
}
}
```

Output: -

```
Enter N: 5
Enter Element of array:
Enter Element 1: 20
Enter Element 2: 25
Enter Element 3: 5
Enter Element 4: 2
Enter Element 5: 3
Enter N which multiple you want to print: 5
20
25
5
```

8. Write a program to print only odd elements from given array elements using for each loop.

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter N:");
int n=sc.nextInt();
int a[]=new int[n];
System.out.println("Enter Element of array:");
for (int i = 0; i < a.length; i++){
        System.out.println("Enter Element "+(i+1)+":");
        a[i]=sc.nextInt();
}
System.out.println("Odd nums:");
for (int val : a){
        if(val%2==1)
        System.out.println(val);
}
}
}
```

Output: -

```
Enter N: 5
Enter Element of array:
Enter Element 1:20
Enter Element 2:23
Enter Element 3:16
Enter Element 4:15
Enter Element 5:87
Odd nums:
23
15
87
```

5.2 Two Dimensional Arrays - Declaration and Initialization: -

Declaration Syntax: -
Data type [][] array_name =new int [row_size][col_size];

```
int [][] arr =new int [3][3] ; //Compile time initialization
arr [0][0]=10;
arr [0][1]=12;
arr [0][2]=15;
arr [1][0]=22;
arr [1][1]=35;
arr [1][2]=47;
arr [2][0]=65;
arr [2][1]=77;
arr [2][2]=25;
```

10 [0][0]	**12** [0][1]	**15** [0][2]
22 [1][0]	**35** [1][1]	**47** [1][2]
65 [2][0]	**77** [2][1]	**25** [2][2]

Example: - (Compile time Initialization)

```
class Create_array  {
public static void main(String[] args) {
int [][] arr =new int [2][2];
arr[0][0]=10;
arr[0][1]=20;
arr[1][0]=30;
arr[1][1]=40;
System.out.println("show elements in array arr");
for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
System.out.println("Elements arr["+ i +"]["+j+"]=" +arr[i][j]);
}
}
}
}
```

Example: - (Run time Initialization)

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int arr[][]=new int[2][2];
System.out.println("Enter Element of array:");
for(int i=0;i<2;i++){
for(int j=0;j<2;j++){
        arr[i][j]=sc.nextInt();
}
}
System.out.println("show elements in array arr");
for(int i=0;i<2;i++){
for(int j=0;j<2;j++){
         System.out.println("Elements arr["+ i +"]["+j+"]=" +arr[i][j]);
}
}
}}
```

Output: -

Enter Element of array: 1 2 3 4

show elements in array arr

Elements arr[0][0]=1

Elements arr[0][1]=2

Elements arr[1][0]=3

Elements arr[1][1]=4

Matrix Operations: -

1. Write a program to print addition of two 2*2 matrix.

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int a[][]=new int[2][2];
int b[][]=new int[2][2];
int c[][]=new int[2][2];
System.out.println("Enter Element of matrix a:");
for(int i=0;i<2;i++){
for(int j=0;j<2;j++){
        a[i][j]=sc.nextInt();
}
}
System.out.println("Enter Element of matrix b:");
for(int i=0;i<2;i++){
for(int j=0;j<2;j++){
        b[i][j]=sc.nextInt();
}
}
for(int i=0;i<2;i++){
for(int j=0;j<2;j++){
        c[i][j]=a[i][j]+b[i][j];
}
}
System.out.println("Resultant Matrix C:");
for(int i=0;i<2;i++){
for(int j=0;j<2;j++){
System.out.print(c[i][j]+"\t");
}
System.out.print("\n");
}
}
}
```

Output: -

```
Enter Element of matrix a:
1
2
3
4
Enter Element of matrix b:
1
2
3
4
Resultant Matrix C:
2	4
6	8
```

2. Write a program to print transpose of a 3*3 matrix.

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
	Scanner sc = new Scanner(System.in);
	int a[][]=new int[3][3];
	int b[][]=new int[3][3];

	System.out.println("Enter Element of matrix:");
	for(int i=0;i<3;i++){
		for(int j=0;j<3;j++){
			a[i][j]=sc.nextInt();
		}
	}
	for(int i=0;i<3;i++){
		for(int j=0;j<3;j++){
		b[j][i]=a[i][j];
		}
	}
	System.out.println("Original Matrix:");
	for(int i=0;i<3;i++){
		for(int j=0;j<3;j++){
		System.out.print(a[i][j]+"\t");
```

```
                }
        System.out.print("\n");
        }

        System.out.println("Transpose Matrix:");
        for(int i=0;i<3;i++){
                for(int j=0;j<3;j++){
                System.out.print(b[i][j]+"\t");
        }
        System.out.print("\n");
        }
        }
}
```

Output: -

```
Enter Element of matrix:
1 2 3 4 5 6 7 8 9
Original Matrix:
1       2       3
4       5       6
7       8       9
Transpose Matrix:
1       4       7
2       5       8
3       6       9
```

3. Write a program to print multiplication of two n*n matrix.

```
import java.util.*;
class Create_array  {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter n:");
int n=sc.nextInt();
int a[][]=new int[n][n];
int b[][]=new int[n][n];
int c[][]=new int[n][n];
System.out.println("Enter Element of matrix a:");
for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
```

```
        a[i][j]=sc.nextInt();
}
}
System.out.println("Enter Element of matrix b:");
for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
        b[i][j]=sc.nextInt();
}
}
for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
c[i][j]=0;
for(int k=0;k<n;k++)
{
        c[i][j]+=a[i][k]*b[k][j];
}
}
}
System.out.println("Resultant matrix c:");
for(int i=0;i<n;i++){
for(int j=0;j<n;j++){
        System.out.print(c[i][j]+"\t");
}
System.out.print("\n");
}
}
}
```

Output: -

```
Enter n: 2
Enter Element of matrix a:
1 2 3 4
Enter Element of matrix b:
1 2 3 4
Resultant matrix c:
7       10
15      22
```

For each loop for 2-D Array: -
Example: -

```
int[][] uu = new int[2][2];
uu[0][0] = 5;
uu[0][1] = 2;
uu[1][0] = 3;
uu[1][1] = 4;
for(int[] u: uu){
                for (int i:u){
                System.out.print(i+" ");
            }
                System.out.println();
}
```

Output: -

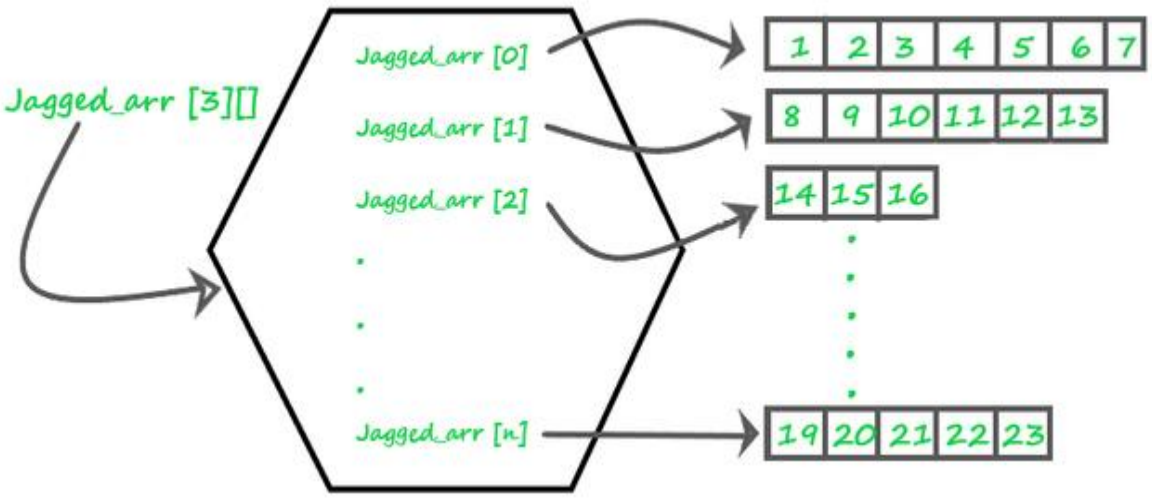
```
5 2
3 4
```

5.3 Ragged Array and Multidimensional Array:-

Ragged array is an array with more than one dimension each dimension has different size.

There is no requirement that all rows in a two-dimensional array have the same length—an array with rows of non-uniform length is known as a ragged array.

A jagged or ragged array is an array of arrays such that member arrays can be of different sizes, i.e., we can create a 2-D array but with a variable number of columns in each row.



1. Declaring a Ragged Array

```
double[][] ir_array = new double [3][];
```

Here, ir_array is a ragged array and the dimension on the right-hand side [3][] specifies one dimension as fixed and another dimension as unknown.

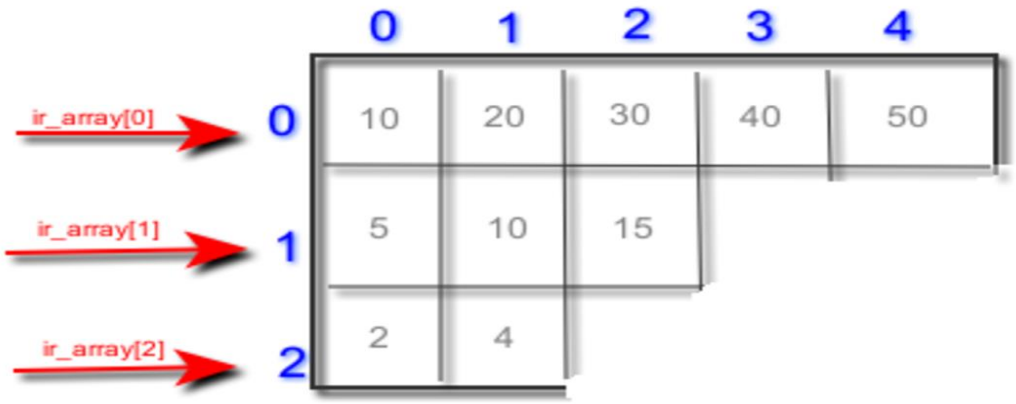
2. Setting the Array References

```
ir_array[0] = new double[5];
ir_array[1] = new double[3];
ir_array[2] = new double[2];
```

3. Populating the Ragged Array

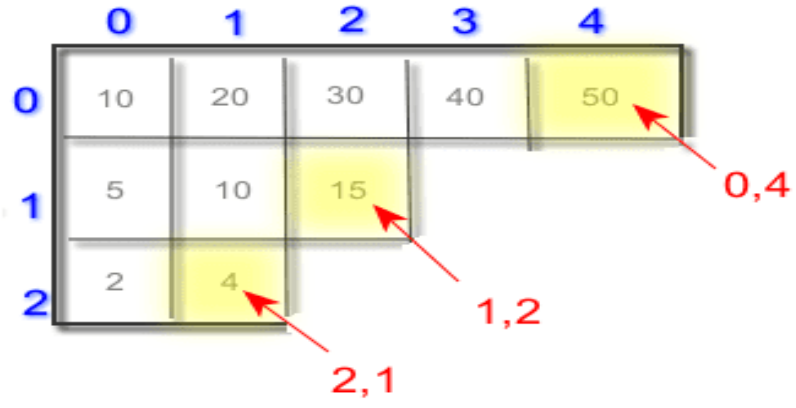
```
for (int i = 0; i < 5; i++) {
    ir_array[0][i] = (i + 1) * 10;
```

```
}
for (int i = 0; i < 3; i++) {
    ir_array[1][i] = (i + 1) * 5;
}
for (int i = 0; i < 2; i++) {
    ir_array[2][i] = (i + 1) * 2;
}
```



4. Picking Ragged Array Element

```
System.out.println("ir_array[0][4] = " + ir_array[0][4]);
System.out.println("ir_array[1][2] = " + ir_array[1][2]);
System.out.println("ir_array[2][1] = " + ir_array[2][1]);
```



Example: -

```
public class Rarray{
        public static void main(String s[]){
        int a[][]=newint[3][];
        a[0]= newint[3];
        a[1]= new int[2];
        a[2]= new int[1];
        a[0][0]=1;
        a[0][1]=2;
        a[0][2]=3;
        a[1][0]=4;
        a[1][1]=5;
        a[2][0]=6;
        for(int i=0;i<a.length;i++)
        {
        for(int j=0;j<a[i].length;j++)
        {
        System.out.println(a[i][j]);
        }
        }
}}
```

Output: -

```
1
2
3
4
5
6
```

Program: - Write a Program to find max elements from given ragged array where number of rows size is 3 and columns are 3,2,1 respectively.

```
import java.util.*;
```

```
class Create_array  {
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int a[][]=new int[3][];
    a[0]= new int[3];
    a[1]= new int[2];
    a[2]= new int[1];

    System.out.println("Enter Element of Ragged Array:");
    for(int i=0;i<a.length;i++){
            for(int j=0;j<a[i].length;j++){
                    a[i][j]=sc.nextInt();
            }
    }
    int max=a[0][0];
    for(int i=0;i<a.length;i++){
            for(int j=0;j<a[i].length;j++){
                    if(a[i][j]>max)
                            max=a[i][j];
            }
    }
    System.out.println("Max:"+max);
    }}
```

Output:-

```
Enter Element of Ragged Array:
1
2
5
44
33
7
Max:44
```