Chapter 8:-

Multi Dimensional Array:

Def: It's like some linear Arrays (one dimension Arrays) under others like a Matrix with Rows and Columns

ex:

```
[0,1,2] -> Arr1
[3,4,5] -> Arr2 ->  ( 0,1,2 )
[6,7,8] -> Arr3     ( 3,4,5 )
                    ( 6,7,8 )
```

and we can delare as the following Syntax:

```
elementType[][] arrayRef;
elemenType arrayRef[][];
```

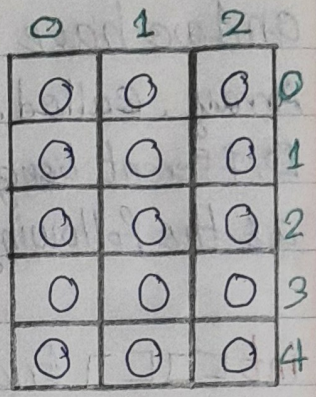
Two Dimensional Array

and you can Create it as following Syntax:

```
elementType[][] arrayRef = new elementtype[1][2];
                                          Rows
                                          Columns
```

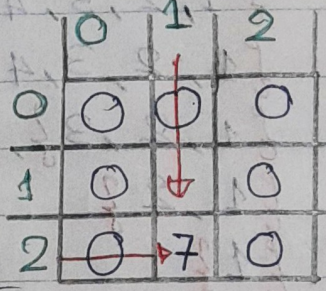
note: The array with any dimension is always based 0

```
int [][] Matrix = new int [5][3];
```



and we can assign a Specific Value in a Specific Place or index as the following Syntax:

```
Matrix [2][1] = 7;
```



and you can make initializing To your Two Dimensional Array as the following Syntax:

```
int [][] Matrix = { { 1, 2, 3 } , { 4, 5, 6 } };
```

Columns (1, 2, 3) — Rows ({1,2,3}, {4,5,6})

in the following example we will explain how to optain the length of two dimensional Array

ex:

```
int [][] matrix = new [3] [4];
int i = 0;
int a = matrix.length;
int b = matrix[i].length;
int c = b - a;
System.out.print(c);
```

elements ← [3] [4] → indecies of each element

a = 3 rows (elements)

b = 4 Columns in row 'i'

4 - 3 = 1

output: 1

and we have another Type of two dimensional Array called Ragged Arrays which each row has Different lengths of columns and we can create it as the following Syntax :

```
int[][] triangleArray = {
    {1,2,3,4,5},
    {1,2,3,4},
    {1,2,3},
    {1,2}
    {1}
};
```

```
int[][] triangleArray = new int[5][];
                        if you don't know the values
triangleArray[0] = new int[5];
                   To intialize the length of columns in row '0'
```

note: in any array you should fill the first index [i][] with Specified number to create an array at least

---

Processing Two-Dimensional Arrays :

we have Some Processes on the Two dimensional Array :

[1] initializing :-

```
1   for(int row = 0 ; row < matrix.length ; row++) {
2     for(int Column = 0 ; Column < matrix[row] ; Column++){
3       matrix [row ] [Column] = input.nextInt();
4     }
5   }
```

note: If you want making Random initialling you should replace line 3 with the following Syntax :

```
.....  The same nested loop above        int Random numbers [0,99]
3   matrix [row ] [column] = (int )(Math.random() * 100) ;
```

note: If you want print elements of the array like rows and Columns you should replace line 3 and line 5 with the following Syntax :

```
.....
3   System.out.print(matrix [row ] [column] + " ");
4   }
5   System.out.println(); }
```

[2] ~~Summing~~ Summing all elements :-

~~int~~

```
int total = 0;
for (int i = 0 ; i < matrix.length ; i++) {
  for(int j = 0; j < matrix[i].length ; j++) {
    total += matrix[i][j] ;
  }
}
```

3 Summing elements by Column :-

```
for(int i = 0 ; i < matrix[i].length ; i++) {
   int total = 0 ;
   for(int j = 0 ; j < matrix.length ; j++) {
      total += matrix[j][i] ;
   }
   System.out.println("Sum for column " + i +
         " is " + total);
}
```

4 Random Shuffling :-

```
for(int i = 0 ;  i < matrix.length ; i++) {
  for(int j = 0 ; j < matrix[i].length ; j++) {
    int i1 = (int)(Math.random() * matrix.length) ;
    int j1 = (int)(Math.random() * matrix[i].length);
    int temp = matrix[i][j];
    matrix[i][j] = matrix[i1][j1];
```

```
        matrix[i1][j1] = temp;
    }
}
```

[5] Row with largest Sum :-

```
int maxRow = 0;
int indexOfmaxRow = 0;
for(int i = 0; i < matrix[i].length; i++){
    maxRow += matrix[0][i];
}
for (int i = 1; i < matrix.length; i++){
    int totalOfthisRow = 0;
    for(int j = 0; j < matrix[i].length; j++){
        totalOfthisRow += matrix[i][j];
        if (totalOfthisRow > maxRow){
            maxRow = totalOfthisRow;
            indexOfmaxRow = i;
        }
    }
}
System.out.println("Row" + indexOfmaxRow +
    "has the maximum sum of " + maxRow);
```

Passing Two-Dimensional Arrays to Methods:

As an array of one dimension you can Pass an Array Two-Dimensional to a method or Return it as a DataType Value or Refvar → Reference variable and you can make it as the following Syntax:

```
public static int[][] methodname(){
}
```
Return DataType

```
public Static DataTypeReturn method(int[][]a){
}
```
Parameter

notes: Remember that when you Passing or Return any Type of arrays you just return it as Preference Variable and Passing it by Value Because it's an Object like Strings

---

Multidimensional Arrays:

Def: This Type of Arrays Consists of an array one dimensional arrays ~~Can be Genera~~ and a three dimensional Array Consists of ~~of~~ an array of two dimensional Arrays and you can write it as the following Syntax:

```
elementType[][][] arrayRefvar=new elementType[i][j][k];
```

ex: Suppose you want to Record Scores of 4 students in 3 exams with Two Types of Questions (MCQ, Essay) you can create a three dimensional array as following Syntax:

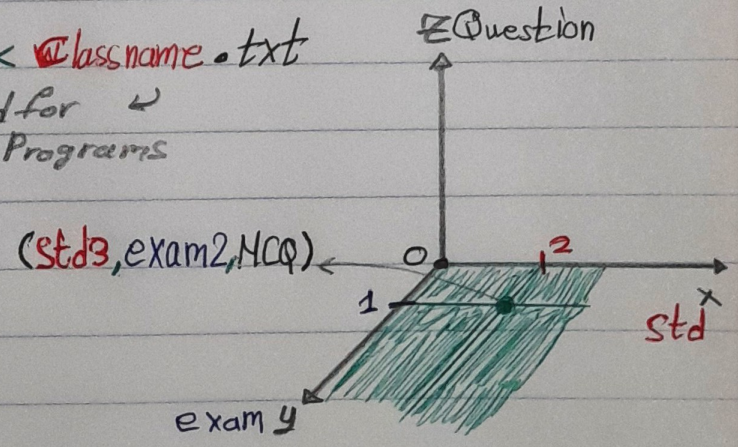
```
double [][][] scores = new double[4][3][2];
```

		exam1		exam2		exam3	
		MCQ (0)	Essay (1)	MCQ (0)	Essay (1)	MCQ (0)	Essay (1)
0	std1	0.0	0.0	0.0	0.0	0.0	0.0
1	std2	0.0	0.0	0.0	0.0	0.0	0.0
2	std3	0.0	0.0	0.0	0.0	0.0	0.0
3	std4	0.0	0.0	0.0	0.0	0.0	0.0

if we want to select std 3 with exam2 Question 1 as an MCQ

```
Scores[2][1][0]
```

you can imagine it as a Space Co-ordinates (X, Y, Z)



note: java classname < classname.txt

Command for run Programs

you can assign a value like following Syntax:

```
Scores[2][1][0] = 9.5;
```