

# Multi-Dimensional Arrays

## ■ Recall:

```
int[] Mark = new int[5];
```

created an array called Mark that could contain 5 integers

Index	0	1	2	3	4
Mark	<int>	<int>	<int>	<int>	<int>

## ■ Problem

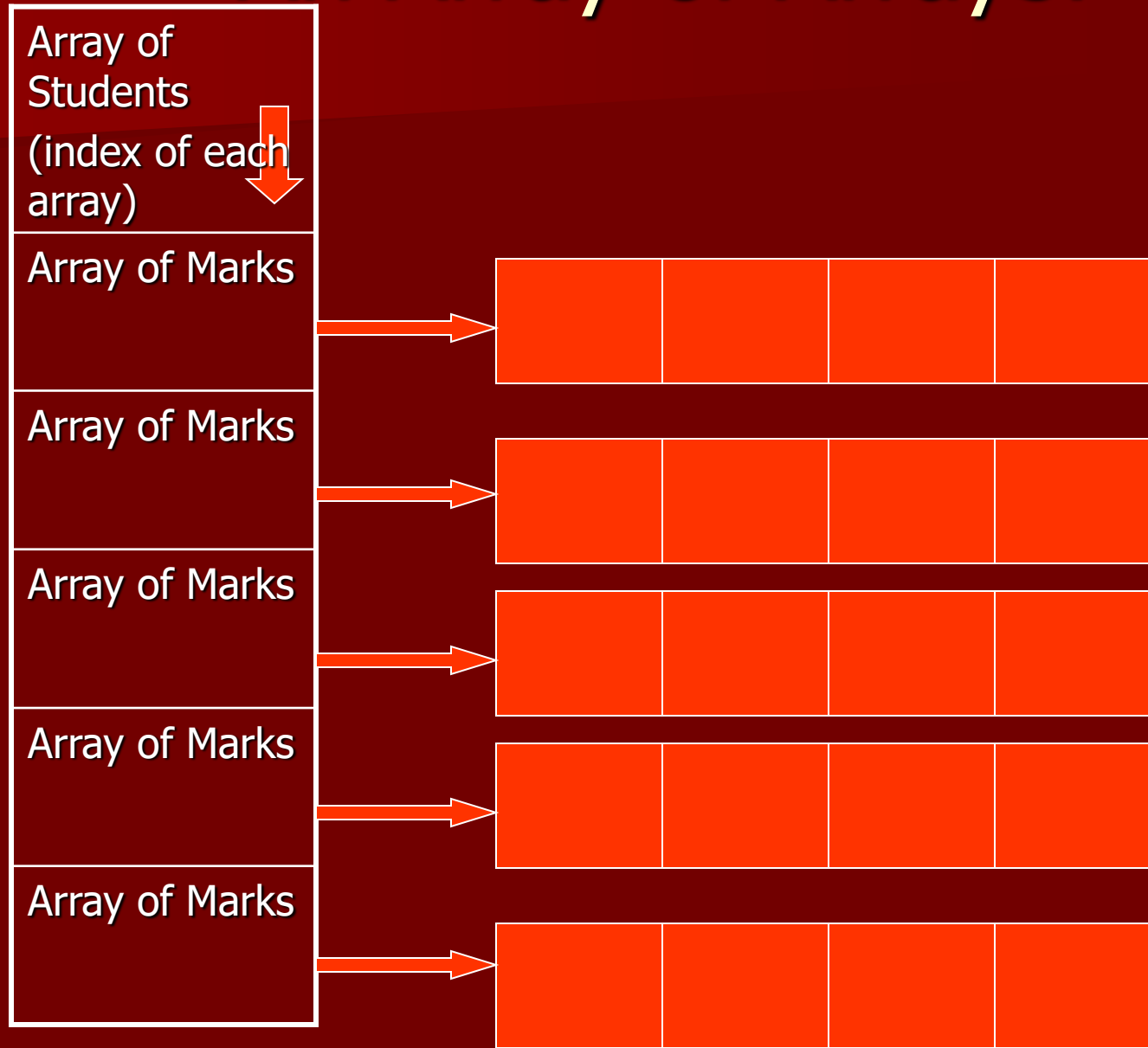
- Create an array of 5 students. For each student create an array of 4 marks each.

## ■ The elements of the array are identified by two indices :

- the first is for the index of each set of data (array)
- the second for the index of the element within each set of data.

## ■ We want to create an array of type arrays!

# An Array of Arrays!



# Declaring a 2-D Array in Java

Notice that the 2-D array is made up of an array of 5 sets of marks (each student is a set), where each set refers to an array of 4 marks each)

Creating our array of arrays:

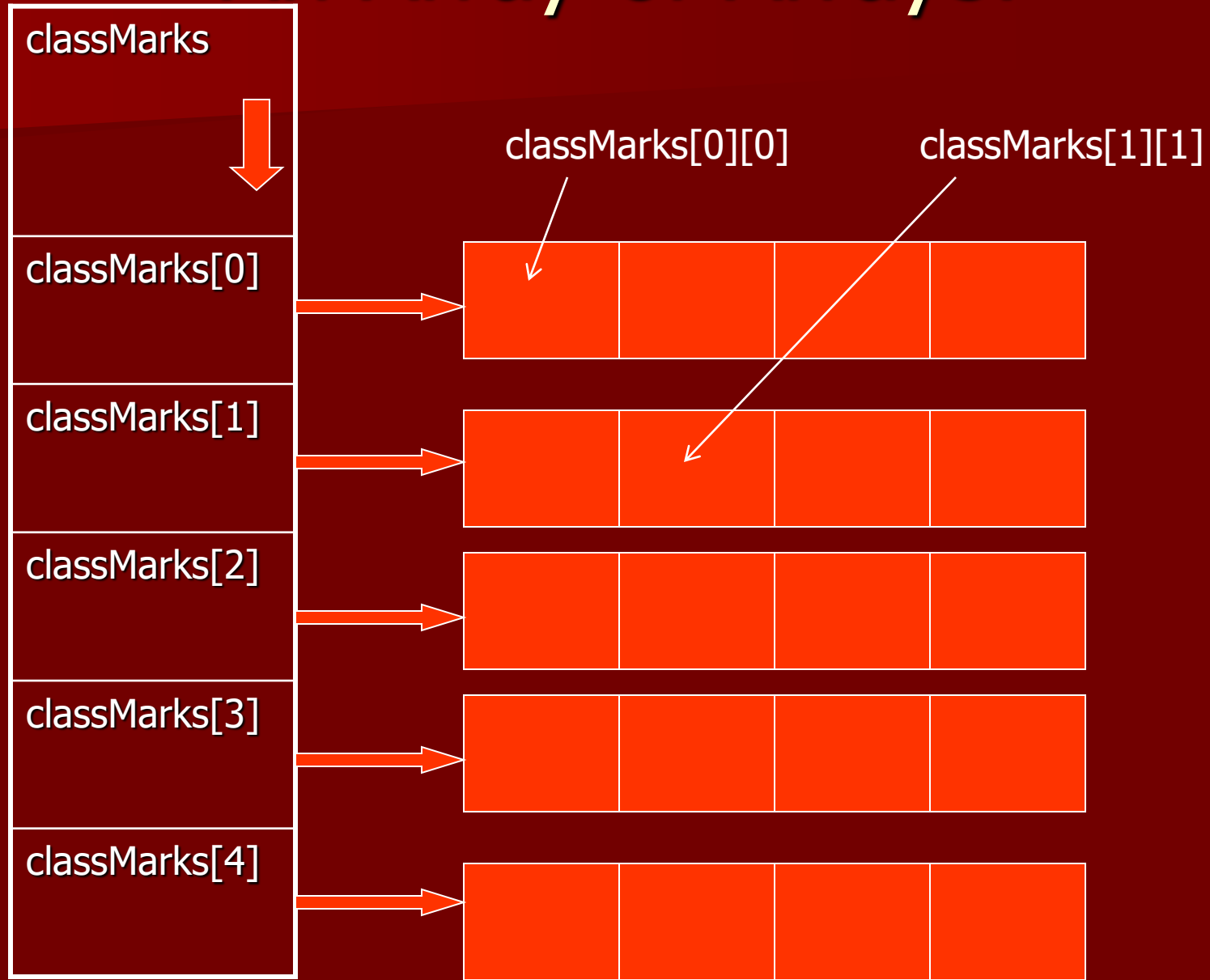
```
int[][] classMarks = new int[5][4];
```



{index of array reference}  
{There are 5 students}

{index of each element}  
{there are 4 courses for each student}

# An Array of Arrays!



# Assigning a value to an element

First identify the index of the element you wish to change

Assign the element a new value

# Where is classMark[3][1]?





# What happens if `classMarks[3][1]=75`?



Using a loop to access  
each element in a 2-D  
Array

- The first loop will loop through each set (student) of data
- The nested loop will loop through the marks for each student

```
for (int student = 0; student < classMark.length; student++) {  
    for (int mark = 0; mark < classMark[0].length; mark++) {  
        System.out.println(classMark[student][mark]);  
    }  
}
```

Note: that the number of students is given by `classMark.length`, while the number of marks in row 0 is given by `classMark[0].length`

OR

We can initialize an array when declaring it!

```
int[][] classMark = {{1,1,1,1},  
                     {1,1,1,1},  
                     {1,1,1,1},  
                     {1,1,1,1},  
                     {1,1,1,1}   };
```

Note: array of 5 arrays of 4 integers each!