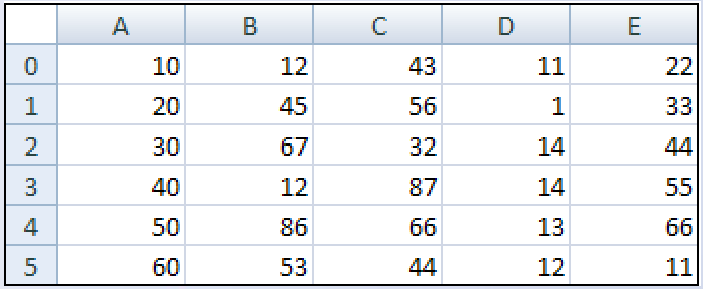
Lecture 6

Multidimensional Arrays in Java

IT TALENTS
Training Camp

Definition

- The arrays you have been using so far have only held one column of data. But you can set up an array to hold more than one column.As an example, think of a spreadsheet with rows and columns.If you have 6 rows and 5 columns then your spreadsheet can hold 30 numbers. It might look like this:



	A	B	C	D	E
0	10	12	43	11	22
1	20	45	56	1	33
2	30	67	32	14	44
3	40	12	87	14	55
4	50	86	66	13	66
5	60	53	44	12	11

IT TALENTS
Training Camp

Definition

- A multidimensional array is an array containing one or more arrays.

- Java can work with multidimensional arrays that are two, three, four, five, or more levels deep.

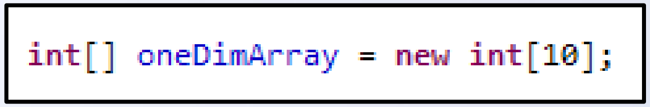
- However, arrays more than three levels deep are hard to manage for most people.

IT TALENTS
Training Camp

Definition and instantiating multidimentional arrays

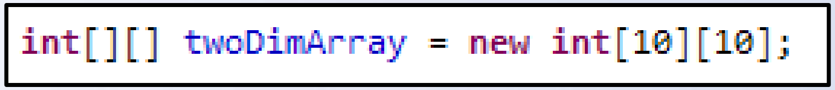
The number of indices determine the dimension of the array.

- 1 dimension – 1 index



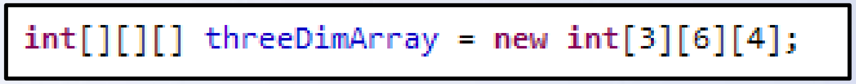
```
int[] oneDimArray = new int[10];
```

- 2 dimensions (also called a matrix) – 2 indices



```
int[][] twoDimArray = new int[10][10];
```

- 3 dimensions – 3 indices and so on ...

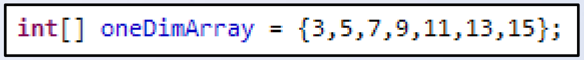


```
int[][][] threeDimArray = new int[3][6][4];
```

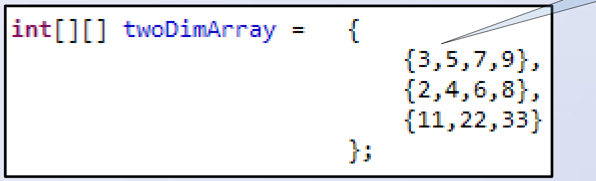
Definition and instantiating multidimentional arrays

- Another way of defining arrays:
  - 1 dimension
  - 2 dimensions
  - 3 dimensions

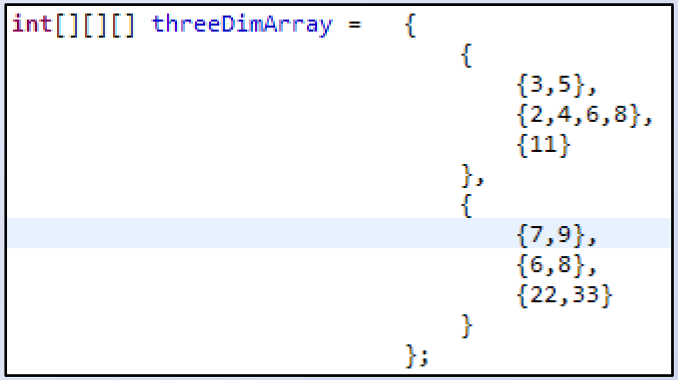
Is this a matrix ?



```
int[] oneDimArray = {3,5,7,9,11,13,15};
```



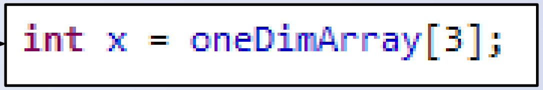
```
int[][] twoDimArray =   {
                            {3,5,7,9},
                            {2,4,6,8},
                            {11,22,33}
                        };
```



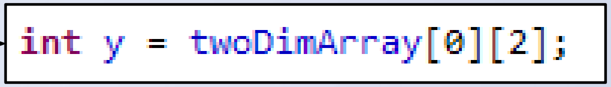
```
int[][][] threeDimArray =   {
                                {
                                    {3,5},
                                    {2,4,6,8},
                                    {11}
                                },
                                {
                                    {7,9},
                                    {6,8},
                                    {22,33}
                                }
                            };
```

Accessing 2D Array's Data

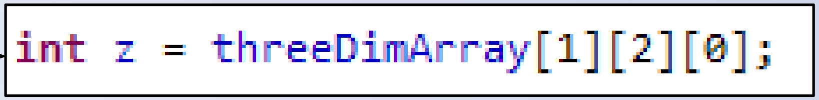
- We access the entries in multidimensional arrays by defining the coordinates of the entry in the array. The number of needed coordinates (indices) depends on the number of dimensions of the array
  - 1 dimension
  - 2 dimensions
  - 3 dimensions



```
int x = oneDimArray[3];
```



```
int y = twoDimArray[0][2];
```



```
int z = threeDimArray[1][2][0];
```

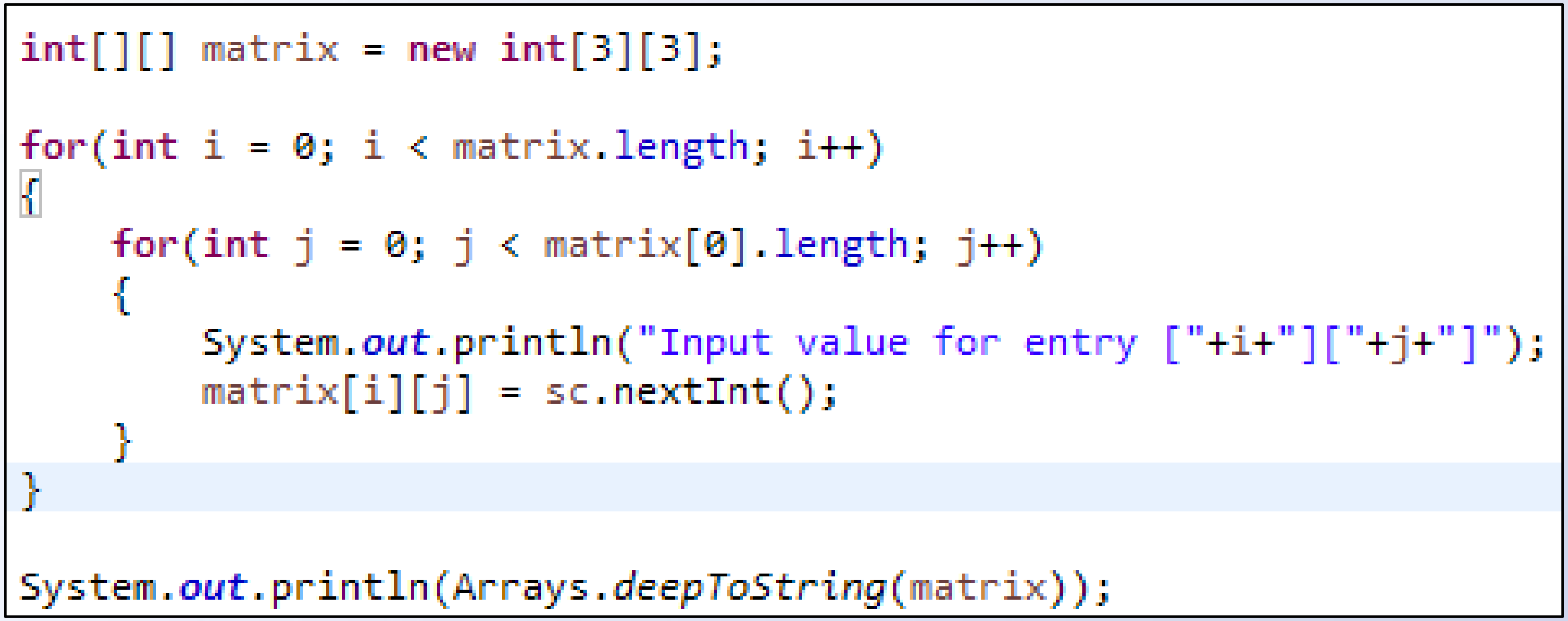
Reading 2D Array

For reading 2D you may use two loops – one for the rows and one for each column in each row. Then in the inner loop you can require a user input for the entry on this row and this column

Try it!

IT TALENTS
Training Camp

Reading 2D Array

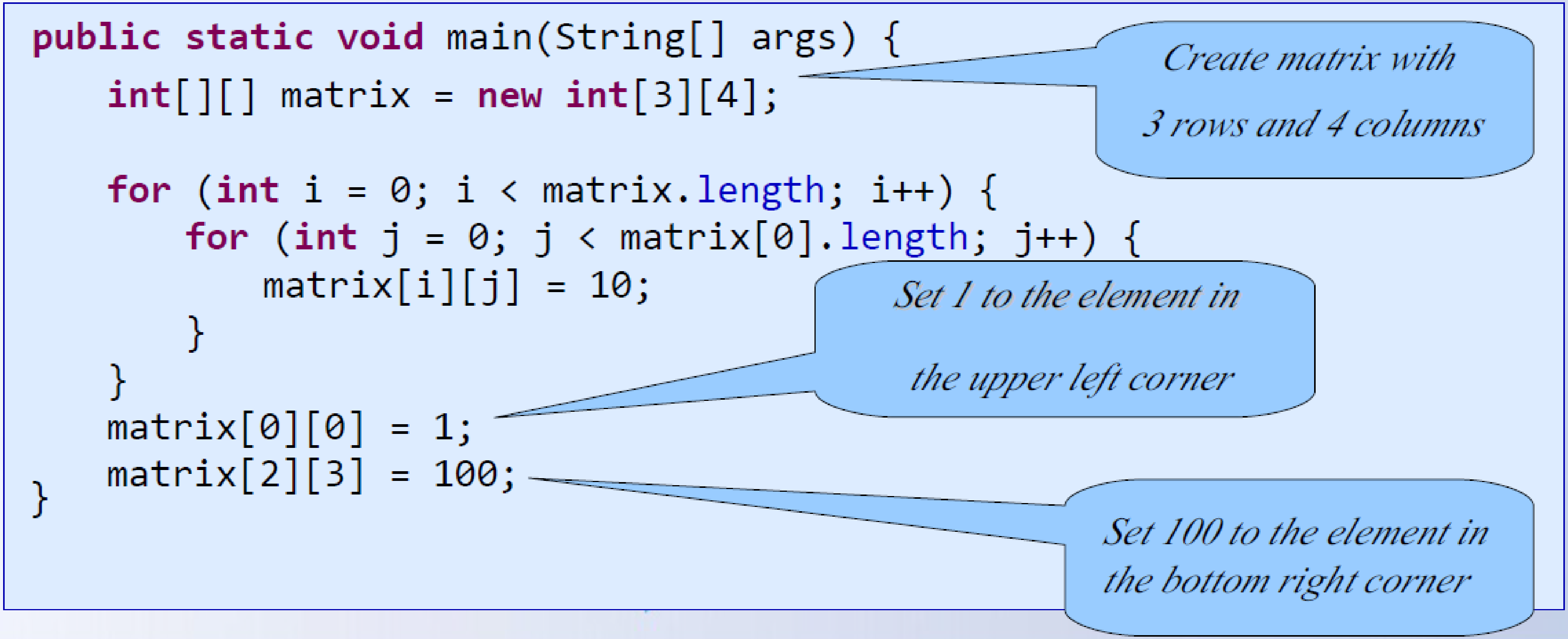


```java
int[][] matrix = new int[3][3];

for(int i = 0; i < matrix.length; i++)
{
    for(int j = 0; j < matrix[0].length; j++)
    {
        System.out.println("Input value for entry ["+i+"]["+j+"]");
        matrix[i][j] = sc.nextInt();
    }
}

System.out.println(Arrays.deepToString(matrix));
```

Accessing 2D Array's Data



```
public static void main(String[] args) {
    int[][] matrix = new int[3][4];

    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[0].length; j++) {
            matrix[i][j] = 10;
        }
    }
    matrix[0][0] = 1;
    matrix[2][3] = 100;
}
```

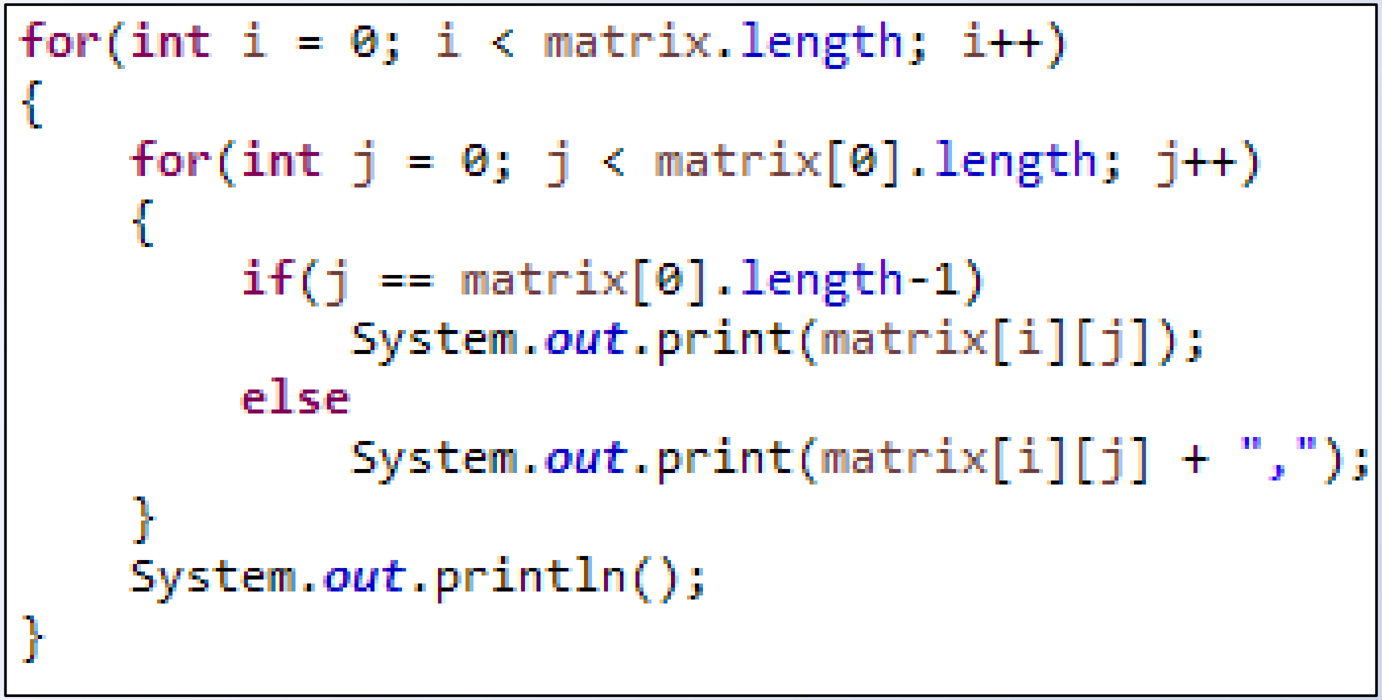
Create matrix with 3 rows and 4 columns

Set 1 to the element in the upper left corner

Set 100 to the element in the bottom right corner

Try to write the above matrix on the console in the form of a table!

Writing a matrix on the console



```
for(int i = 0; i < matrix.length; i++)
{
    for(int j = 0; j < matrix[0].length; j++)
    {
        if(j == matrix[0].length-1)
            System.out.print(matrix[i][j]);
        else
            System.out.print(matrix[i][j] + ",");
    }
    System.out.println();
}
```