*CSE 102*

**ARRAYS-2**

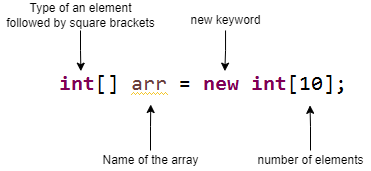
A Quick Recap:

* An array is a sequence of fixed-length data items each having the same type and stored contiguously in memory.
* Once created, the size of the array cannot be changed.
* The name (identifier) of an array is a reference to its first element.
* We use arrays to store data which will be necessary in the next stages of our computation.
* If the type of an array is T[] where T is a primitive type, each element of that array is a value of type T.

If the type of an array is T[][] where T is a primitive-type, each element of that array is a reference to another array whose elements are values of type T.

T[][][] and further can be defined in a similar way.

Java Syntax for Arrays:



Here is a 2 dimensional 3 by 5 array (an array of references to 3 integer arrays each having size 5). Note that each element of each array is initialized to the default value (0 for int).

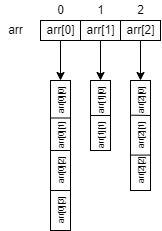


If you want to initialize your array with different values, you can use the following syntax:



In case of multi-dimensional arrays:





**Exercises**:

1. What will be the output of the following program?

**public** **static** **void** main(String[] args){

**int**[][] arr = **new** **int**[3][2];

System.***out***.println(arr[2][1]);

}

1. 1
2. 3
3. 0
4. Runtime Error
5. What will be the output of the following program?

**public** **static** **void** main(String[] args){

**int**[][] arr = {{1, 2}, {4, 5, 2}};

System.***out***.println(arr[0][2]);

}

1. 0
2. Runtime Error
3. Compiler Error
4. 2
5. What will be the output of the following program?

**public** **static** **void** main(String[] args){

**int**[][] arr = **new** **int**[3][3];

arr[0] = **new** **int**[5];

System.***out***.println(arr[0][4]);

}

1. 5
2. 0
3. Runtime Error
4. 3
5. The following function is supposed to print the elements of ith integer array as ith row to standard output. What should be replaced with question marks?

**static** **void** printArray(**int**[][] arr) {

**for**(**int** i=0; i<arr.length; i++) {

**for**(**int** j=0; j< ??? ; j++)

System.***out***.print(arr[i][j] + " ");

System.***out***.println();

}

}

1. arr.length
2. arr.length / 2
3. arr[j].length
4. arr[i].length
5. What will be the output of the following program? *printArray* is the function defined in the previous question.

**public** **static** **void** main(String[] args){

**int**[][] arr = {{2,2,2,2}, {4,4,4}};

arr[0] = arr[1];

*printArray*(arr);

}

1. 2 2 2 2

4 4 4

1. 2 2 2 2

2 2 2 2

1. 4 4 4

4 4 4

1. 0 0 0

4 4 4

1. What will be the output of the following program? *printArray* is the same as the one in previous question.

**public** **static** **void** main(String[] args){

**int**[][] arr = {{2,2,2,2}, {4,4,4}};

arr[0] = arr[1];

arr[0][0] = 1;

*printArray*(arr);

}

1. 1 0 0

4 4 4

1. 1 4 4

4 4 4

1. 4 4 4

4 4 4

1. 1 4 4

1 4 4

1. The following method is supposed to return the jth column of *arr* as an integer array. Which statement should be written in place of the comment?

**static** **int**[] getColumn(**int**[][] arr, **int** j) {

**int**[] res = **new** **int**[arr.length];

**for**(**int** i=0; i<arr.length; i++)

// Here

**return** res;

}

1. res[i] = arr[i][j];
2. res[j] = arr[i][j];
3. res[i] = arr[j][i];
4. res[j] = arr[j][i];
5. Write a function which takes a 2 dimensional array of strings and returns true iff the array is rectangular.
6. Write a function which takes a 2 dimensional array of chars “*words”* and returns a 1 dimensional array of strings whose ith element is the string built from the ithchar array of *words.*

**Definition**: Let us call a 2-d array *arr* a ***matrix (or rectangular array)***if for all i,j < arr.length : arr[i].length = arr[j].length.

**Definition**: We define the **size**(or **dimensions**) of a matrix *m* as **height x width** where height = *m*.length and width = *m*[0].length. Note that changing the 0 to any other index within the range leads to an equivalent definition because of the way we defined matrix before.

**Definition**: A matrix *m* of type T[][] is called **non-empty** if m.length > 0 and m[i] is not null for any i if T is a non-primitive type.

1. Write a function which takes two non-empty double matricesthat have the same dimensions and returns their sum, which is again a double matrix having the same dimensions. (Do not alter the original matrices, create a new one).
2. Write a function which takes a non-empty integer matrix and returns its maximum element.
3. Write a function which takes a non-empty Boolean matrix and returns its transpose.

**ANSWERS:**

1. C
2. B
3. B
4. D
5. C
6. D
7. A
8. **static** **boolean** isRect(String[][] a) {

**for**(**int** i=1; i<a.length; i++)

**if**(a[i].length != a[0].length)

**return** **false**;

**return** **true**;

}

1. **static** String[] getStrings(**char**[][] words) {

String[] res = **new** String[words.length];

**for**(**int** i=0; i<words.length; i++) {

res[i] = "";

**for**(**int** j=0; j<words[i].length; j++)

res[i] += words[i][j];

}

**return** res;

}

1. **static** **double**[][] matrixSum(**double**[][] a, **double**[][] b) {

**double**[][] res = **new** **double**[a.length][a[0].length];

**for**(**int** i=0; i<a.length; i++)

**for**(**int** j=0; j<a[0].length; j++)

res[i][j] = a[i][j] + b[i][j];

**return** res;

}

1. **static** **int** matrixMax(**int**[][] m) {

**int** max = m[0][0];

**for**(**int** i=0; i<m.length; i++)

**for**(**int** j=0; j<m[0].length; j++)

**if**(m[i][j] > max)

max = m[i][j];

**return** max;

}

1. **static** **boolean**[][] transpose(**boolean**[][] m){

// swap the dimensions

**boolean**[][] res = **new** **boolean**[m[0].length][m.length];

**for**(**int** i=0; i<m.length; i++)

**for**(**int** j=0; j<m[0].length; j++)

res[j][i] = m[i][j];

**return** res;

}