

Computer Science AP

Two Dimensional Arrays (Matrices)

We've learned how to create an array. An array can be thought of as a list in memory. The line `int[] A = new int[5]` might read as, 'make an array of length 5 using integers'. This could be visualized as

0
0
0
0
0

with each index position containing a value (zero by default for integers). But what if, instead of a value, you had an array at each position in the array? To create 'an array of arrays' is done with a line like this: `int[][] A = new int[5][3]`. This might read as, 'Make an array of length 5 using arrays of length 3 using integers'. It could be visualized like as

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

I've sketch the above to look like a rectangular grid that is 3 wide and 5 tall, but it would also be perfectly fine to visualize this in your head as a grid that is 5 wide and 3 tall, since the actual 'shape' of this data structure in memory is not actually a rectangle! Who's to say that arrays should be drawn on paper up and down versus left to right, or even as rectangles?

Once you create the 2D array (which I'll refer to as a matrix or grid sometimes), it's easy to use. Here are some of the basics.

If you want to know the dimensions of an array you are given, you can use:
`int a = A.length;` //this would be 5 for our example above
`int b = A[0].length;` //this would be 3 for our example above

To set the value at a certain position in the grid you can just do this
`A[0][0] = 1;`
`A[4][2] = 99;`
`A[2][0] = 55;`

`A[0][2] = 33;`

to give the following (remember that index 0 is the first index in both arrays!)

1	0	33
0	0	0
55	0	0
0	0	0
0	0	99

To set the 'far bottom corner' of a matrix of unknown dimensions to 1234 then you could use

`A[A.length-1][A[0].length-1] = 1234;`

or for this example

`A[4][2] = 1234;`

Example – Movie Theater

A theater has 30 rows of seats and each row contains 40 seats. Just to clarify, you can have 40 people in the 'first row' of the theater.

The theater allows people to pre-book their seats by naming the row and column in the theater. The theater must be able to check whether a seat is booked or not.

As part of some software, you are asked to do the following:

1. Create a 2D array of booleans that represents the seats in the theater.
2. Set all the seats to empty by setting their value to false. (true means taken)
3. Printing out the array.

Try to do on your own now, but if not...

Solution at end of this document.

You must convince yourself that this 'loop in a loop' works nicely for going through all the seats in the grid. It is a very common grid routine.

Note: USUALLY the the preferred way to write code for a 2d array is

`[column][row]` or `[x][y]` when representing physical items in the real world that are grid like. But also note that there is no rule about doing this!

Example – Movie Theater Continued

1. Someone phones in to the movie theater and says that they want to book the entire first row of the theater (all 40 seats).
If the entire first row is currently available, then set the entire first row to book.

*Try this on your own now, but if not
Solution at the end of this document.*

2. Ask the user to enter a row and column of a seat in the theater and check to see if it is available. Tell them if it is or not.

*Try this on your own now, but if not
Solution at the end of this document.*

Now you have a basic idea how to create and use a 2d array, you're ready for some real problems to challenge your brain.

Movie Theater Example Solution

```
boolean[][] seats = new booleans[40][30];
for (int row=0; row<seats.length; row++) {
    for (int col=0; col<seats[0].length; col++) {
        seats[col][row] = false;
    }
}

for (int row=0; row<seats.length; row++) {
    for (int col=0; col<seats[0].length; col++) {
        System.out.print( seats[col][row] );    //notice just .print is used, not println
    }
    System.out.println(""); //go to the next line
}
```

//is the first row available? Check entire first row. If you find one seat already taken,
//then they are not all free.

```
boolean allFree=true;
for (int col=0; col<seats[0].length; col++) {
    if (seats[col][0] == true) {
        allFree = false;
    }
}
```

```
//book the first row
for (int col=0; col<seats[0].length; col++)
    seats[col][0] = true;
```

```
//check to see if a certain seat is available
int col = //valid user input
int row = //valid user input
if (seats[col][row]==false)
    System.out.println("Available!");
else
    System.out.println("Not Available :( ");
```