

AP CS: Lesson 15.4: Multi-Dimensional Arrays

Name: _____ Period: _____

Java Syntax:

2D Arrays:

2D Array Creation and use:

```
type[][] name = new type[height][width];
type[][] name = { {value00, value01, ..., value0w},
                  {value10, value11, ..., value1w},
                  ...
                  {valueh0, valueh1, ..., valuehw} }
// initializing an array explicitly
name [indexh][indexw] = value; // Assigning a value to an array's indexed location
public static type methodName(type[][] name) { // using an array as a parameter
public static type[][] methodName(parameters) { // returning an array
methodName(arrayName); // calling a method with an array as its parameter
type[][] name = methodName(parameters); // assigning an array as a returned result
```

2D Array Examples:

```
char[][] board = new char[3][3]; // creates an array of chars 3 by 3, 9 elements
char[][] board = {{'X', 'O', 'X'},
                  {'O', 'X', 'O'},
                  {'X', 'O', 'X'}};
public static void PlayGame(char[][] input) {} // a char 2D array as a parameter
public static boolean[][] emptyLoc(int x, int y) {} // returning a 2D boolean array
printBoard(board); // calling a method with a 2D array as its parameter
boolean[] values = emptyLoc(x, y); // assigning a 2D array as a returned result
```

Jagged 2D Array Examples:

```
int[][] jagged = new int[3][];
jagged [0] = new int[2];
jagged [1] = new int[5];
jagged [2] = new int[4];
```

2D Array Length Field:

nums.length – returns array's height or number of rows (no [])
nums[0].length – returns array's width or columns - at that row [], which is the same for regular arrays.

Examples: for a 2D Array:

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

Length results would be:

```
nums.length is 5
nums[0].length is 4
```

All rows same width here. Note: these would differ for a Jagged Array

```
nums[1].length is 4
nums[2].length is 4...
```

Acting on 2D arrays usually involves nested for loops using the length dimensions – keep track of row & columns carefully and name variables wisely.

```
public static void fillArray (int nums[][]){
    int count = 1;
    for (int row = 0; row < nums.length ; row++){
        for (int col = 0; col < nums[0].length ; col++){
            <statements...>
        }
    }
}
```