Chapter 8

Multidimensional Arrays
Multi-dimensional arrays
A two-dimensional can be viewed logically as a table or a matrix, with each element referenced by a row and col. More precisely, by the intersection of a row and col, like in any spreadsheet.
In reality however, the array is more like an array of arrays. In memory the array is a linear structure, with each row having the same number of columns.

Note that the compiler needs to know how many columns make up a row in order to access the elements properly. In the above case, every two columns make up one row, so to access the second row, all it has to do is skip 2 elements, 4 to access the third row and so on.
2D arrays have both rows and columns

```java
string[][] fullName = new String[STUDENT_COUNT][2];

double[][] citySalesTax = new double[STATE_COUNT][CITY_COUNT];

int[][] studentScores = new int[STUDENT_COUNT][ASSESSMENT_COUNT];
```
A two-dimensional array for recording the daily lo and hi temperatures for 2002, for the county of Madison, Illinois.

```java
int[][] dailyLoHiTempForMadison = new int[365][2];
int[][] dailyLoHiTempForMadison = new int[2][365];
```

A two-dimensional array for recording the annual average rainfall for all 50 states for the last 10 years.

```java
int[][] annualAverageRainfallForUSStates = new int[50][10];
int[][] annualAverageRainfallForUSStates = new int[10][50];
```
Initializing 2D arrays

// Use initializer list to assign values in row
// first order.
int[][] mat = { {1, 2},
              {3, 4},
              {5, 6}
  };

<table>
<thead>
<tr>
<th>[0]</th>
<th>[1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0]</td>
<td>1</td>
</tr>
<tr>
<td>[1]</td>
<td>3</td>
</tr>
<tr>
<td>[2]</td>
<td>5</td>
</tr>
</tbody>
</table>
int[][] mat = new int[2][3];

// The outer for represents the rows.
// The inner the columns.
for (int r = 0; r <= 1; r++) {
    for (int c = 0; c <= 2; c++) {
        mat[r][c] = r * 3 + c + 1;
    }
} // end for
} // end for

To allow a row first processing order, which is C++'s way of doing things, the outer for must represent the rows, while the inner for the columns.
public void readNamesFromKeyboard(String[][] name) {
    // Ask the user for name.length names.
    for (int row = 0; row <= name.length - 1; row++) {
        System.out.println("Enter name [First Last]: ");
        for (int col = 0; col <= name[row].length - 1; col++) {
            name[row][col] = keyboard.next();
        }
    }
} // end readNamesFromKeyboard()
public void writeNamesToScreen(String[][] name) {

    // Display each name to the screen, one per line.
    for (int row = 0; row <= name.length - 1; row++) {
        for (int col = 0; col <= name[row].length - 1; col++) {
            System.out.println(name[row][col]);
        } // end for
    } // end for

} // end writeNamesToScreen()