2D Arrays - Reading

Arrays store only one list of values, such as a list of scores for a student. If we wanted to store the scores for five students we would need to create a new array for each student.

An easier way to do this would be to create a **2D array**. A 2D array has **two dimensions** and is very similar to cells in a spreadsheet or table.

The following table shows the data stored in a 2D Array named s*cores*.

	0	1	2	3	4
0	6	9	10	3	7
1	4	7	10	5	5
2	5	8	9	4	7

Notice that we do not store any column names as the array is used to store the data itself and not any titles. The numbers 0 to 4 at the top and 0 to 2 at the left show the **references** for the columns and rows. Just as with **1D arrays**, the column or row reference starts at 0.

To assign to a 2D array use:
arrayName ← `[[array1],[array2],…]`

```
scores ← [[6, 9, 10, 3, 7], [4,
7, 10, 5, 5], [5, 8, 9, 4, 7]]
```

To reference a 2D array use:
***arrayName*[*rowNumber*] [*columnNumber*]**

```
scores[2][1]

#This will evaluate to 8 as this
is the value in row 2 column 1.
```

Let's say that the final test was incorrectly marked and all the scores for the final test/row should be increased by 1 mark. This is now easy to do because we can use a **FOR loop**.

```
FOR i ← 0 TO 4
    scores[2,i] ← scores[2,i] + 1
ENDFOR
```

If we now needed to convert all the scores to percentages by multiplying each number by 10 we can use a nested FOR loop to do this.

```
FOR i ← 0 TO 2
    FOR j ← 0 TO 4
        scores[i,j] ← scores[i,j] * 10
    ENDFOR
ENDFOR
```

In programming we usually give **variable** names meaningful names such as *playerName* or *playerAge*. When we are using array references we usually just use the letter *i* as the variable name for the **counter** in the FOR loop. A nested loop would then use the letter *j* as the counter.

It is possible to have strings or characters stored in an array or 2D array. For example, the following code will place a king at position [0,5] on a chess board.

```
chessBoard[0,5] ← 'King'
```