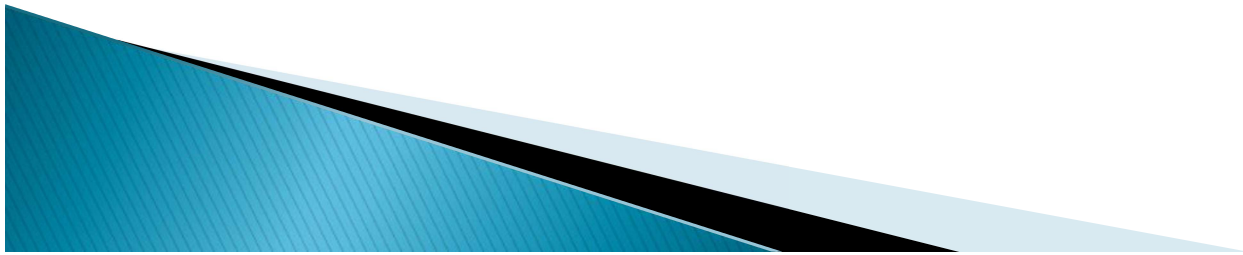


# Outline

- ▶ What are 2D Arrays?
- ▶ Declaration
- ▶ Initialization
- ▶ Examples



# Introduction to 2-D Arrays

- ▶ A 2-D array is a contiguous collection of variables of the same type, that may be viewed as a table consisting of rows and columns.

	Column			
	0	1	2	3
Row 0				
1				
2				

table [1][3]

- ▶ The same reason that necessitated the use of 1-D arrays can be extended to 2-D and other multi-D Arrays.
- ▶ For example, to store the grades of 30 students, in 5 courses require multiple 1-D arrays.
- ▶ A 2-D array allows all these grades to be handled using a single variable.

# Declaration of 2-D Arrays

- ▶ A 2-D array variable is declared by specifying the type of elements, the name of the variable, followed by the number of rows and number of columns – each is a separate bracket:
- ▶ The following declares a 2-D array, *table*, having 3 rows and 4 columns.

```
int table[3][4];
```

- ▶ Both rows and columns are indexed from zero. So the three rows have indexes 0, 1 and 2 and four the columns have 0, 1, 2, 3.
- ▶ As we saw in 1-D array, it is a good practice to declare the sizes as constants. Here is the declaration for the array *table*.

```
#define ROWS 3
```

```
#define COLS 4
```

```
int table[ROWS][COLS];
```

		Column			
		0	1	2	3
Row	0				
	1				
	2				

→ table [1][3]

# Accessing 2-D Array elements

- ▶ A particular element of a 2-D array, table, is referenced by specifying its row and column indexes:

`table[RowIndex][columnIndex]`

- ▶ For example, given the declaration:

```
int table[3][4];
```

- ▶ The following stores 64 in the cell with row index 1, column index 3.

`table[1][3] = 64;`

		Column			
		0	1	2	3
Row	0				
	1				64
	2				

- ▶ We use the same format to refer to an element in an expression:

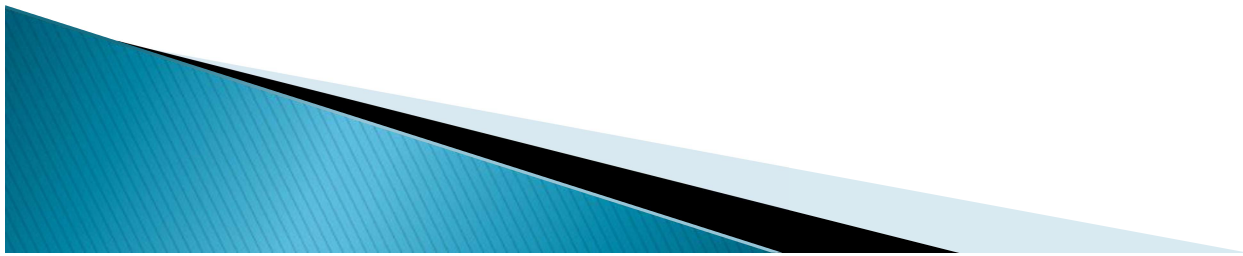
`table[2][3] = table[1][3] + 2;`

		Column			
		0	1	2	3
Row	0				
	1				64
	2				66

# Scanning 2D Arrays

- ▶ To use 2-dimensional array we use Two nested for-loop as follows:

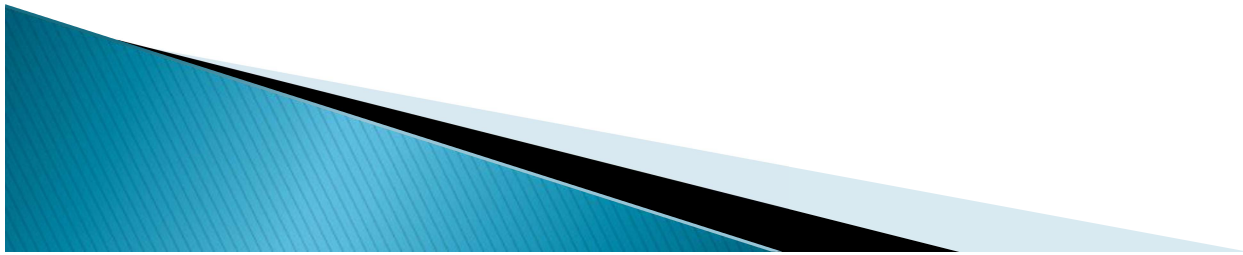
```
for(i=0; i<3;i++)  
    for(j=0; j<3; j++)  
        A[ i ] [ j ] = value ;
```



# Example

## Addition of two matrices

- ▶ Write a program that reads two matrices of size  $5 \times 5$ , and adds the two matrices to another one which will be displayed.



# Example: Stores

## Problem Analysis Chart for Array Problem

Given Data	Required Results
<ol style="list-style-type: none"><li>1. 15 stores</li><li>2. Sales for all stores for seven days</li></ol>	<ol style="list-style-type: none"><li>1. Total sales for each store for the week</li><li>2. Total sales for the company for each day</li><li>3. Total sales for the company for the week</li></ol>
Processing Required	Solution Alternatives
<ol style="list-style-type: none"><li>1. Summing store sales</li><li>2. Summing day's sales</li><li>3. Summing all sales</li></ol>	<ol style="list-style-type: none"><li>1. Use two-dimensional array for data, one-dimensional array for store totals and day's totals, and a single variable for grand total</li><li>2. Use a two-dimensional array to store all values</li></ol>



# Solve Stores Problem using 2D Arrays in Class

