

# Two Dimensional Arrays

# Matrix

35

- ▶ Set of number arranged in rows and columns
- ▶ Numbers are called elements of matrix
- ▶ An Image is nothing but a matrix
- ▶ If we say  $m \times n$  matrix, then that means  $m$  rows and  $n$  columns
- ▶  $2 \times 3$  matrix:

$$\begin{bmatrix} 1 & 3 & 8 \\ 2 & -4 & 5 \end{bmatrix}$$

# 2D Array

- ▶ An Array inside another array
- ▶ Just like a table/matrix where there are rows and columns

36

		Columns				
		0	1	2	3	4
Rows	0	Alan 00	Bob 01	Carol 02	David 03	Ellen 04
	1	Fred 10	Grace 11	Henry 12	Ian 13	Jen 14
	2	Kelly 20	Liam 21	Mary 22	Nancy 23	Owen 24

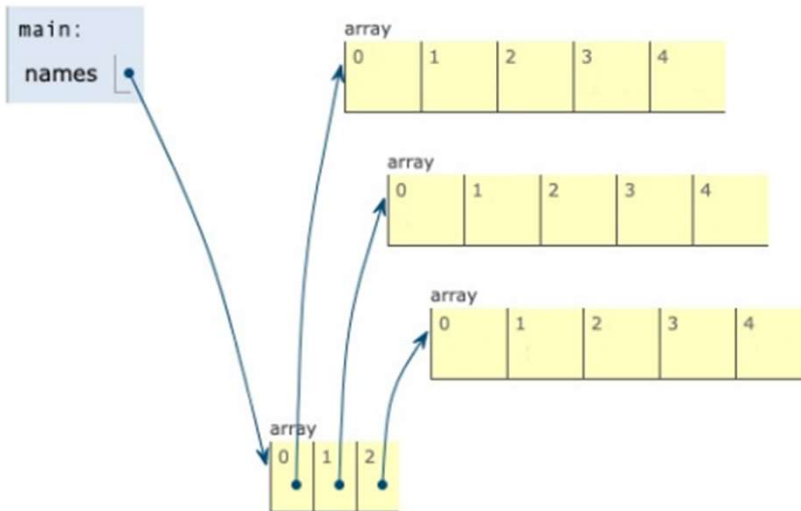
### 2D Array Syntax

- Array type followed by a name for the 2D array followed by **two** empty pairs of brackets [].
- The number of **rows** goes inside the **first** pair of brackets and the number of **columns** goes inside the **second** pair of brackets.

```
string names[3][5];
```

# 2D Array Syntax

# Visualization of 2D Array



```
string names[3][5];  
  
cout << sizeof(names) / sizeof(names[0]) << " rows" << endl;  
cout << sizeof(names[0]) / sizeof(string) << " columns" << endl;
```

Using sizeof() with 2D Array to find number of rows and columns

```
string names[ ][5] = { {"Alan", "Bob", "Carol", "David",  
    "Ellen"},  
    {"Fred", "Grace", "Henry", "Ian", "Jen"},  
    {"Kelly", "Liam", "Mary", "Nancy",  
    "Owen"} };  
  
cout << names[1][2] << endl;
```

## Initializing and Accessing 2D Arrays

```
string names[3][5] = { {"Alan", "Bob", "Carol", "David",  
                        "Ellen"},  
                      {"Fred", "Grace", "Henry", "Ian", "Jen"},  
                      {"Kelly", "Liam", "Mary", "Nancy",  
                        "Owen"} };  
  
cout << names[1][2] << endl;  
  
names[1][2] = "Harry";  
cout << names[1][2] << endl;
```

## Update/Modification in 2D Arrays



```
int digits[3][3] = { {1, 2, 3},  
                     {4, 5, 6},  
                     {7, 8, 9} };  
  
int row = sizeof(digits) / sizeof(digits[0]); //number of rows  
int col = sizeof(digits[0]) / sizeof(int); // number of columns  
  
for (int i = 0; i < row; i++) {  
    for (int j = 0; j < col; j++) {  
        cout << digits[i][j] << endl;  
    }  
}
```

## Traversing 2D Array using for loop

## Traversing 2D Array using enhanced for loop

```
// Driver code
int main()
{
    const int i = 3, j = 3;
    int matrix[i][j] = {{1, 2, 3},
                        {4, 5, 6},
                        {7, 8, 9}};
    for (auto &row : matrix)
    {
        for (auto &column : row)
        {
            cout << column << " ";
        }
        cout << endl;
    }
}
```

# Task

44

- Write C++ code to sum the even numbers in the following array:

11	15	20
23	24	26
10	20	33

# Future topics

45



ARRAYS AND  
FUNCTIONS



ARRAYS AND  
POINTERS