Two Dimensional Array

- **Two (Multi-dimensional) Dimensional Array –**

	Column 0	Column 1	Column 2
Row 0	x[0][0]	x[0][1]	x[0][2]
Row 1	x[1][0]	x[1][1]	x[1][2]
Row 2	x[2][0]	x[2][1]	x[2][2]

The multidimensional array is also known as rectangular arrays in C++. It can be two dimensional or three dimensional. The data is stored in tabular form (row * column) which is also known as matrix.

Syntax - **data type array-name[size][size];**

Note:

- ***2D arrays can be defined as an array of arrays.***
- ***It can also represent a Matrix.***

- **Initializing a 2D array in C++ -**

Type – I

```
int arr[4][2] = {
{1234, 56},
{1212, 33},
{1434, 80},
{1312, 78}
} ;
```

Type – II

```
int arr[4][2] = {1234, 56, 1212, 33, 1434, 80, 1312, 78};
```

▪ Printing a 2D Array in C++ -

```
#include<iostream.h>
void main( )
{
   int arr[4][2] = {
      { 10, 11 },
      { 20, 21 },
      { 30, 31 },
      { 40, 41 }
      } ;

   int i,j;

   cout<<"Printing a 2D Array:\n";
   for(i=0;i<4;i++)
   {
      for(j=0;j<2;j++)
      {
         cout<<"\t"<<arr[i][j];
      }
      cout<<endl;
   }
}
```

▪ Taking 2D Array Elements As User Input -

```
#include<iostream>
using namespace std;
main( )
{
      int  s[2][2];
      int  i, j;
      cout<<"\n2D Array Input:\n";
      for(i=0;i<2;i++)
      {
        for(j=0;j<2;j++)
        {
            cout<<"\ns["<<i<<"]["<<j<<"]=  ";
            cin>>s[i][j];
        }
      }

      cout<<"\nThe 2-D Array is:\n";
      for(i=0;i<2;i++)
      {
        for(j=0;j<2;j++)
        {
            cout<<"\t"<<s[i][j];
        }
        cout<<endl;
      }
}
```