



# Programming In C++

Course 2: Lecture 3, Arrays

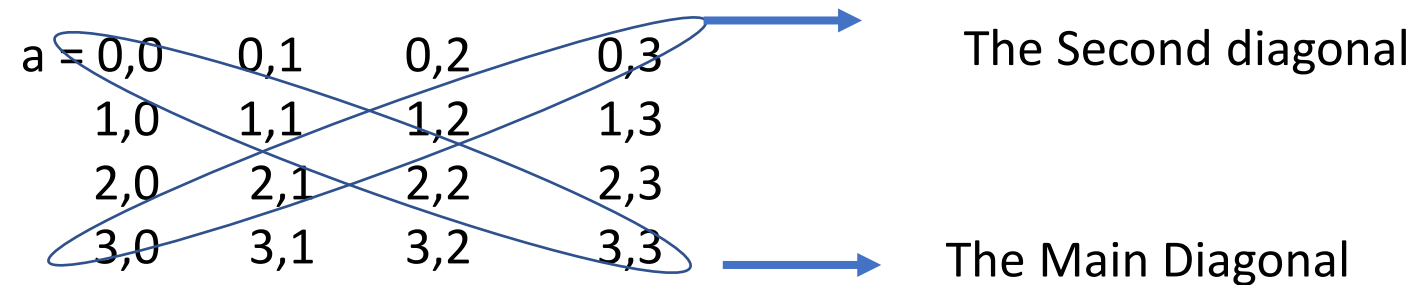
Prepared By Dr. Ali Al-Sabaawi

Nineveh University- Faculty of IT- department of software

# Square Array

The square array is a 2-D array when no. of rows equal no. of columns.

double int a[4][4];



# Square Array

Program to access the main and second diagonal

```
double int a[4][4];
```

a = 0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

```
for(i=0 ; i<4 ; i++)  
    cout<<a[i][i]
```



Access the elements of the main  
diagonal

```
for(i=0, j=3 ; i<4 , j>=0; i++, j--)  
    cout<<a[i][j]
```



Access the elements of the  
second diagonal

# Square Array

Program to access the elements above and below the main diagonal

```
double int a[4][4];
```

a = 0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

```
for(i=0 ; i<4 ; i++)  
for(j=0 ; j<4 ; j++)  
if( i>j)  
cout<<a[i][j]
```

```
for(i=0 ; i<4 ; i++)  
for(j=0 ; j<4 ; j++)  
if( i< j)  
cout<<a[i][j]
```

→ Access the elements above the main diagonal

→ Access the elements below the second diagonal

# Square Array

Program to access the elements below and above the second diagonal

```
double int a[4][4];
```

a = 0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

```
for(i=0 ; i<4 ; i++)  
for(j=0 ; j<4 ; j++)  
if( i+ j< 3)  
cout<<a[i][j]
```

→ Access the elements above the second diagonal

```
for(i=0 ; i<4 ; i++)  
for(j=0 ; j<4 ; j++)  
if( i+ j> 3)  
cout<<a[i][j]
```

→ Access the elements below the second diagonal



# 2-D array

## Convert 2-D to 1-D

```
int inp[3][2] = {{8, 5}, {7, 9}, {6, 3}};  
int out [6];
```

```
int k = 0;  
for (int i = 0; i < 3; i++)  
    for (int j = 0; j < 2; j++)  
        out [k++] = inp[i][j];
```



## Convert 1-D to 2-D

Ex: convert 1-D to 2-D with the size 4\*3

```
int inp [12];  
int output [4][3];
```

```
int k = 0;  
for (int i = 0; i < 4; i++)  
    for (int j = 0; j < 3; j++)  
        output[i][j] = inp[k++];
```

# 2-D array

Find the summation of the boundaries of the 4\*4 array.

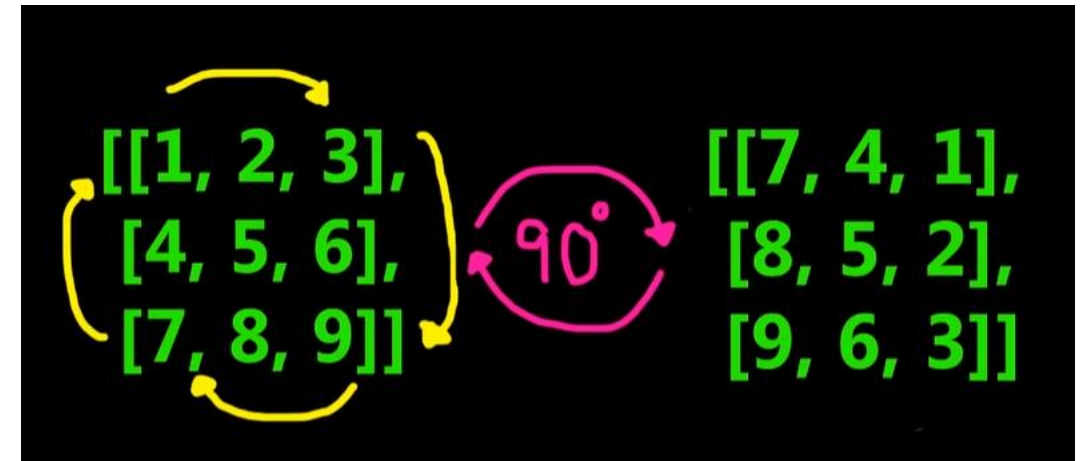
```
for( i=0; i<4 ; i++)  
{ s+=arr[0][i];      → Sum of the first row  
  s+=arr[3][i];      → Sum of the last row  
}
```

```
for( i=0; i<4 ; i++)  
{ s+=arr[i][0];      → Sum of the first column  
  s+=arr[i][3];      → Sum of the last column  
}
```

# 2-D array

Rotate 2-D 90 degrees clockwise. Here the rows become columns. The first row becomes last column, the second row becomes second column and the last row becomes first column

```
For(int i=0 ; i<3 ; i++)  
For(int j=0 ; j<3 ; j++)  
B[ j ][ 2-i ]=a[i][j];
```







# 2-D array



Rotate 2-D 90 degrees anticlockwise. Here the rows become columns. The first row becomes last column, the second row becomes second column and the last row becomes first column

```
for(int i=0 ; i<3 ; i++)  
for(int j=0 ; j<3 ; j++)  
b[ 2-j ][i]= arr[i][j];
```

```
a=[[1, 2, 3],  
   [4, 5, 6],  
   [7, 8, 9]]
```

```
a=[[3, 6, 9],  
   [2, 5, 8],  
   [1, 4, 7]]
```



# The End



