



# ECAP770

## ADVANCE DATA STRUCTURES

---

**Ashwani Kumar**  
Assistant Professor

# Learning Outcomes



After this lecture, you will be able to

- arrays
- one dimension array
- two dimension array

# Array

- An array is variable that can store multiple values.
- Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type.
- Array is non-primitive data type. (int, float, char etc.)
- Arrays consist of contiguous memory location.

# Syntax

```
datatype  
name_of_array[size];
```

Datatype

Name of array

Size (integer)

# Array Initializing and Memory Representation of an Array

- `int arr[5]={ 10,11,12,13,14};`

- `arr`

0	1	2	3	4	index
10	11	12	13	14	
<code>arr[0]</code>	<code>arr[1]</code>	<code>arr[2]</code>	<code>arr[3]</code>	<code>arr[4]</code>	

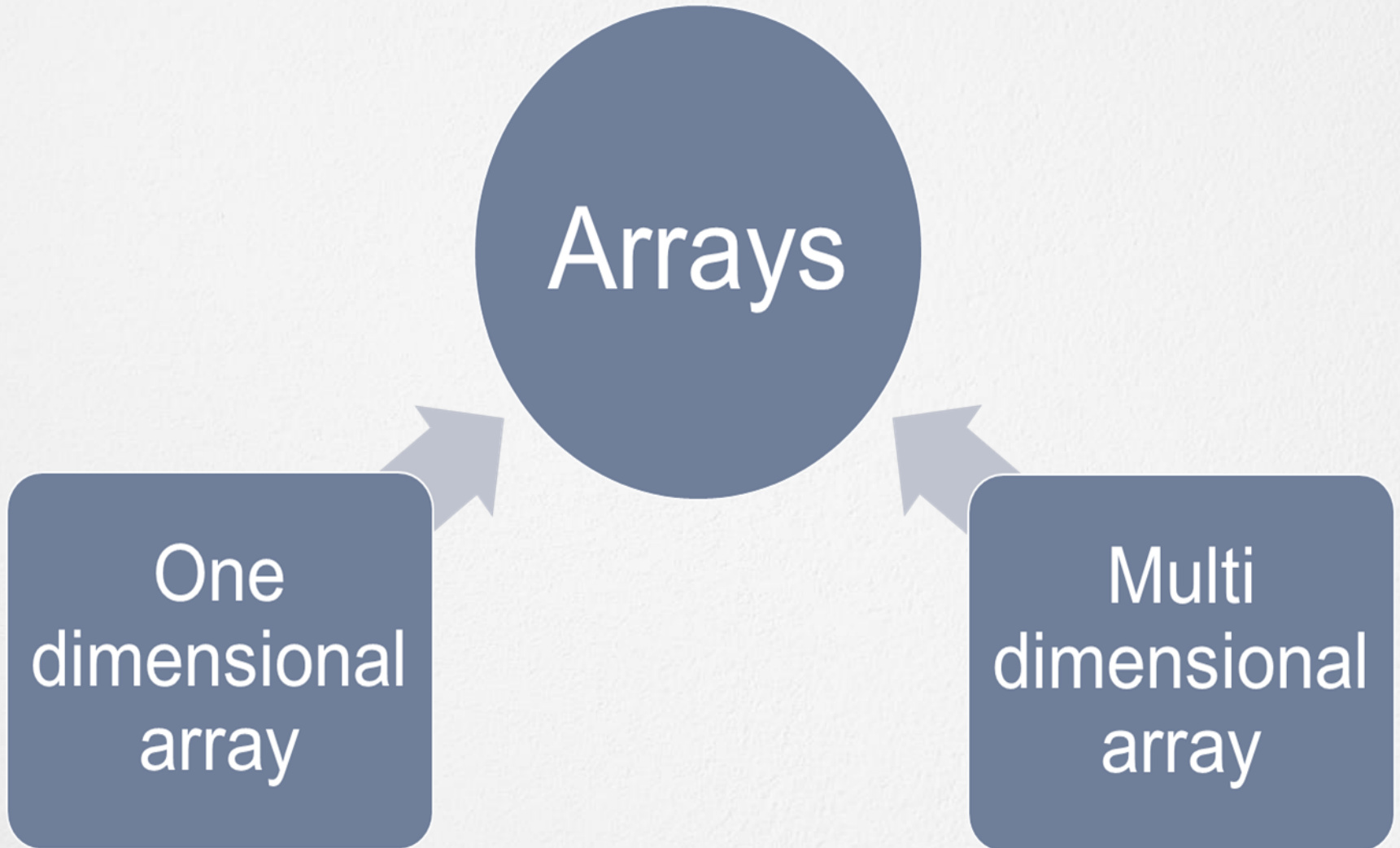
# Access Array Elements

- Array elements access by indices.
- Suppose you declare an array `arr` as above then first element is `arr[0]`, second element is `arr[1]` and so on.





# Type of Array



# Type of Array

- One dimensional Array

```
int a[5];
```

- Multi-dimensional Array

```
int m[3][3];
```



# Two Dimensional Array in C

- An array of arrays is known as two dimensional array.
- Two dimensional array in C language organized as matrices which is used in C programming to represent rows and columns.

# Declaration of Two Dimensional Array in C

- `data_type name_of_array[rows][columns];`
- Datatype
- Name of array
- rows (integer)
- columns (integer)
- Number of elements in 2D arrays is multiply of rows and columns for eg. `Int x[3][3]` has  $3*3$  i.e. 9 elements.

# Array Initializing and Memory Representation of an Array

- `int x[3][4] = {{1,2,3}, {4,5,6}, {7,8,9}};`

<code>a[0][0]=1</code>	<code>a[0][1]=2</code>	<code>a[0][2]=3</code>
<code>a[1][0]=4</code>	<code>a[1][1]=5</code>	<code>a[1][2]=6</code>
<code>a[2][0]=7</code>	<code>a[2][1]=8</code>	<code>a[2][2]=9</code>



Index



Value



Index



Value



Index



Value

# Advantages of 2D Arrays

- 2D Arrays represent multiple data items of the same type in the form of matrix.
- We can perform all matrices operation, sum of matrix, transpose of matrix using 2D Array.

# Advantages of Arrays

- Arrays represent multiple data items of the same type using a single name.
- Elements of array can be accessed randomly by using the index number.
- Easy access to all the elements.
- Sorting and searching becomes easy.

# Applications of Array

- Arrays can be used for CPU scheduling.
- Arrays can be used for reverse data elements, sort data elements etc.
- Arrays are used to Perform Matrix Operations
- Arrays are used to Store List of values
- Arrays are also used to implement stack and queues.



# Program: One Dimension Array

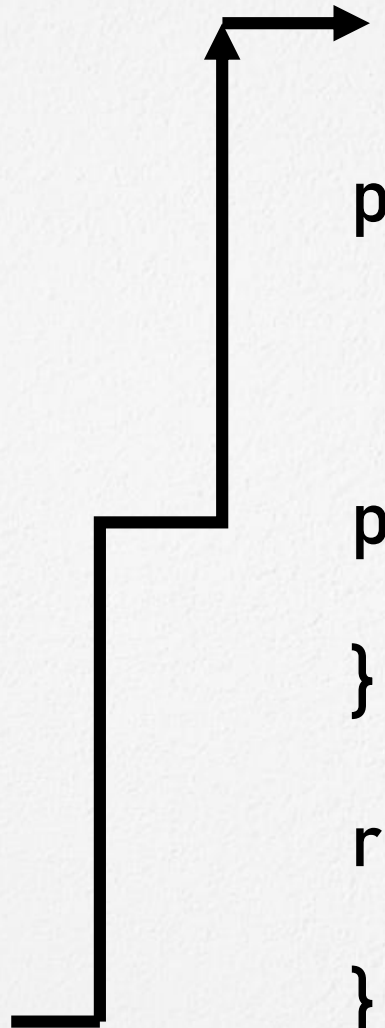
```
#include <stdio.h>

int main()
{
    int arr[5]={10,11,12,13,14};
    for(int i=0;i<=4;i++)
    {
        printf("%d\n",arr[i]);
    }
    return 0;
}
```

# Program: Two Dimension Array

```
#include <stdio.h>

int main()
{
    int x[3][4] = {{1,2,3},
                   {4,5,6}, {7,8,9}};
    int i,j;
    for( i=0;i<=2;i++)
    {
        for(j=0;j<=2;j++)
            {
                printf("%d\t",x[i][j]);
            }
        printf("\n");
    }
    return 0;
}
```



The diagram illustrates the flow of execution for the nested loops. A horizontal line from the innermost loop's closing brace (the last closing brace of the program) extends to the right and then turns upwards as an arrow pointing to the opening brace of the innermost loop, indicating a jump back to the start of the loop iteration.



That's all for now...