# CSE101-Lec

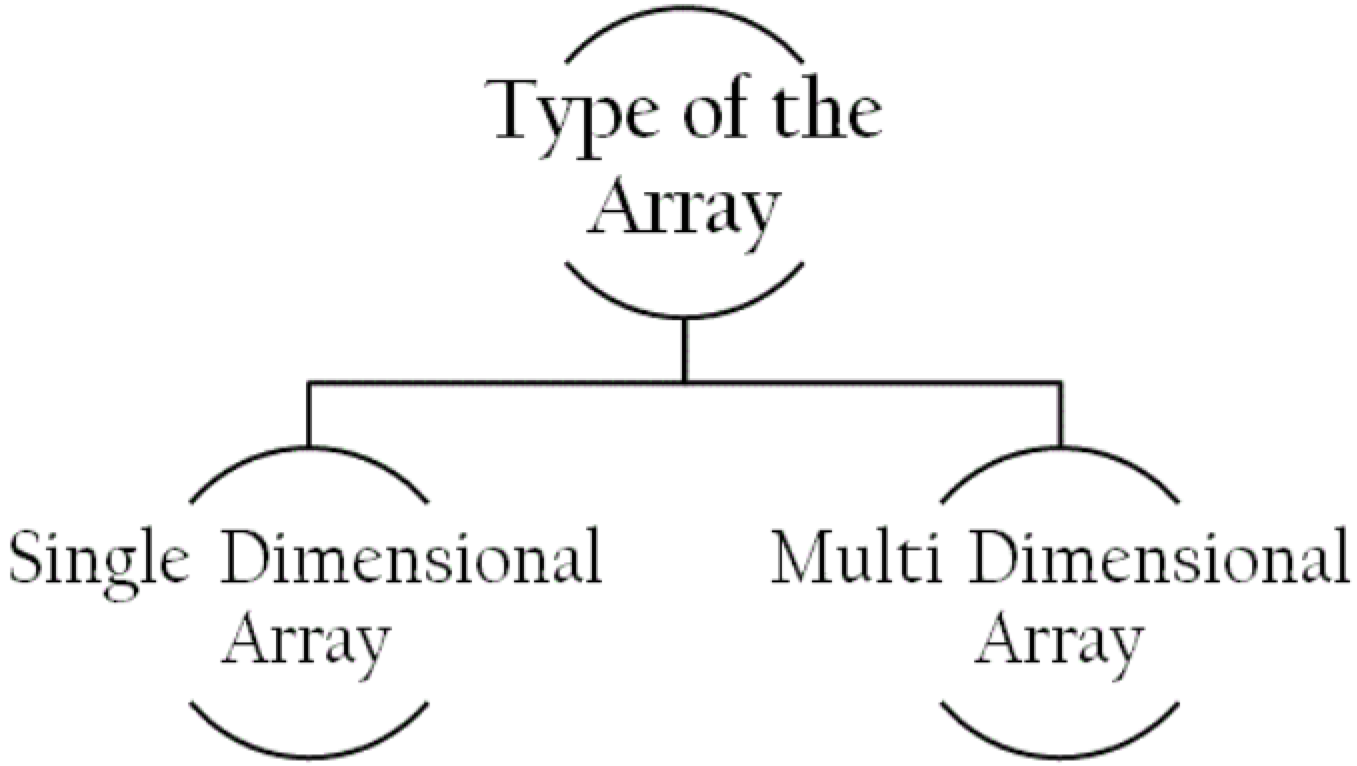
Types of Arrays

Application of arrays

# Outline

- Defining and processing array types
  - 1D array
  - 2D array
- Applications of arrays

## Array Types


Type of the Array
Single Dimensional Array
Multi Dimensional Array

## Single Dimensional array

- Used to represent and store data in a linear form.
- Array having only one subscript variable is called **One-Dimensional array.**
- It is also called as **Linear Array** or **1- D Array.**
- Example: **int** a[n];

```
char arr1[20] = “C Programming" ;
float arr2[3] = {12.5,13.5,14.5};
```

## Program to display the average of elements in 1D array

```
#include<stdio.h>
int main()
{
int avg, sum=0, i;
int marks[30];       /*array declaration*/
for(i=0;i<=29;i++)
{
 printf("enter numbers");
 scanf("%d",&marks[i]);  /*store data in array*/
}
for(i=0;i<=29;i++)
{
  sum = sum + marks[i];/*read data from an array*/
}
avg = sum/30;
printf("Average marks= %d", avg);
}
```

# Multi-Dimensional Arrays

- Array having more than one subscript variable is called **Multi- Dimensional array.**
- Multi Dimensional Array is also called as **Matrix**.
- Most popular and commonly used multi dimensional array is **two dimensional array**.
- The 2-D arrays are used to store data in the form of table.
- Syntax for declaring a two dimensional array:

*datatype arrayName [ rowSize ] [ columnSize ] ;*

- Example:

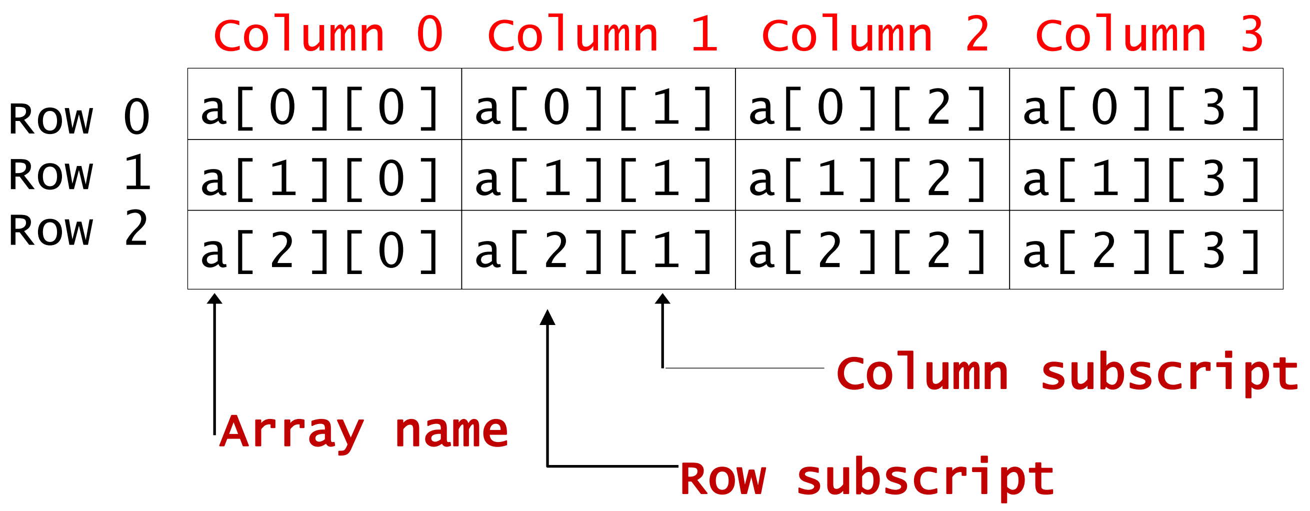
```
int matrix_A [2][3] = { {1, 2, 3},{4, 5, 6} } ;
```

It reserves 6 continuous memory locations of 2 bytes each in the form of 2 rows and 3 columns. And the first row is initialized with values 1, 2 & 3 and second row is initialized with values 4, 5 & 6.

It can also initialize as :

```
int matrix_A [2][3] = { {1, 2, 3},
                        {4, 5, 6} } ;
```

# Multi-Dimensional Arrays



| | Column 0 | Column 1 | Column 2 | Column 3 |
|---|---|---|---|---|
| Row 0 | a[0][0] | a[0][1] | a[0][2] | a[0][3] |
| Row 1 | a[1][0] | a[1][1] | a[1][2] | a[1][3] |
| Row 2 | a[2][0] | a[2][1] | a[2][2] | a[2][3] |

# Multiple-Subscripted Arrays

- **Initialization**
  - int a[ 2 ][ 2 ] = { { 1, 2 }, { 3, 4 } };

| | |
|---|---|
| 1 | 2 |
| 3 | 4 |

  - Initializers grouped by row in braces .
  - If not enough, unspecified elements set to zero

    int a[ 2 ][ 2 ] = { { 1 }, { 3, 4 } };

- **Referencing elements:** via 2 subscripts, i.e. one for row 2nd for column.

| | |
|---|---|
| 1 | 0 |
| 3 | 4 |

  - Specify row, then column

    printf( "%d", a[ 0 ][ 1 ] );

# Program to display 2D array

```c
#include<stdio.h>
void main()
{
int a[3][3], i, j;
for(i=0; i<3; i++) //for loop for rows
{
   for(j=0; j<3;j++) // for loop for columns
   {
        printf("enter the value of a[%d][%d]: ", i, j);
        scanf("%d", &a[i][j]);
   } //end for columns
} //end for rows
printf("elements of 2D matrix are\n");
for(i=0; i<3; i++)
{
 for(j=0;j<3;j++)
   {
   printf("%d\t", a[i][j]);
   }    //end for
 printf("\n");
} //end for
} //end main
```

```
enter the value of a[0][0] :1
enter the value of a[0][1] :2
enter the value of a[0][2] :3
enter the value of a[1][0] :4
enter the value of a[1][1] :5
enter the value of a[1][2] :6
enter the value of a[2][0] :7
enter the value of a[2][1] :8
enter the value of a[2][2] :9
Element of 2D matrix are:
1   2   3
4   5   6
7   8   9
```

## Operations on arrays

- Insertion of element into an array
- Deletion of element from an array

## Program to insert an element into an array

```
#include<stdio.h>
#include<conio.h>

int main()
{
   int array[100], position, c, n, value;
   printf("Enter number of elements in array:\n");
                  scanf("%d", &n);
   printf("Enter %d elements:\n", n);
                                       for (c = 0; c < n; c++)
                                        {  scanf("%d", &array[c]);  }
   printf("Enter the location where to insert an element:\n");
                  scanf("%d", &position);
   printf("Enter the value to insert:\n");
                  scanf("%d", &value);

   for (c = n - 1; c >= position - 1; c--)
   {   array[c+1] = array[c]; }
   array[position-1] = value;

   printf("Resultant array is:\n");
   for (c = 0; c <= n; c++)
   {
printf("%d\n", array[c]);
 }
}
```

```
Enter number of elements in array: 4
Enter 4 elements
2
45
66
33
Enter the location where you wish to insert
an element : 2
Enter the value to insert : 99
Resultant array is :
2
99
45
66
33
```

## Program to delete an element from an array

```
#include <stdio.h>
int main()
{
   int array[100], position, c, n;
   printf("Enter number of elements in array\n");
   scanf("%d", &n);
   printf("Enter %d elements\n", n);
              for (c = 0; c < n; c++)
              {  scanf("%d", &array[c]); }
   printf("Enter the location where you wish to delete from an
array\n");
   scanf("%d", &position);
   for (c = position-1; c < n; c++)
     { array[c] = array[c+1]; }

   printf("Resultant array is\n");
   for (c = 0; c < n-1; c++)
     { printf("%d\n", array[c]); }
}
```

```
Enter number of elements in array: 4
Enter 4 elements
2
45
66
33
Enter the location where you wish to delete from an array : 2
Resultant array is :
2
66
33
```

# Applications of Array

**1. Stores Elements of Same Data Type**

- Array is used to store the number of elements that are of same data type.

  Eg: int students[30];

- Array of marks of five subjects for single student.

  float marks[5];

- Array of marks of five subjects for 30 students.

  float marks[30][5]

- Similarly, if we declare the character array then it can hold only character.
- So, in short character array can store character variables while floating array stores only floating numbers.

## 2. Array Used for Maintaining multiple variable names using single name

Suppose we need to store 5 roll numbers of students, then without declaration of array we need to declare following -

```
int roll1,roll2,roll3,roll4,roll5;
```

1. Now in order to get roll number of first student we need to access roll1.
2. Guess if we need to store roll numbers of 100 students, then what will be the procedure.
3. Maintaining all the variables and remembering all these things is very difficult.

➢ So we are using array which can store multiple values and we have to remember just single variable name.

## 3. Array Can be Used for Sorting Elements

- We can store elements to be sorted in an array and then by using different sorting technique we can sort the elements.

Different Sorting Techniques are :

1. Bubble Sort
2. Insertion Sort
3. Selection Sort

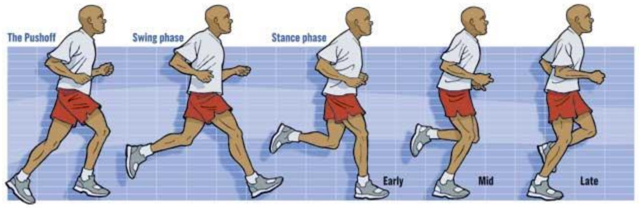
## 4. Array Can Perform Matrix Operation

Matrix operations can be performed using the array. We can use 2-D array

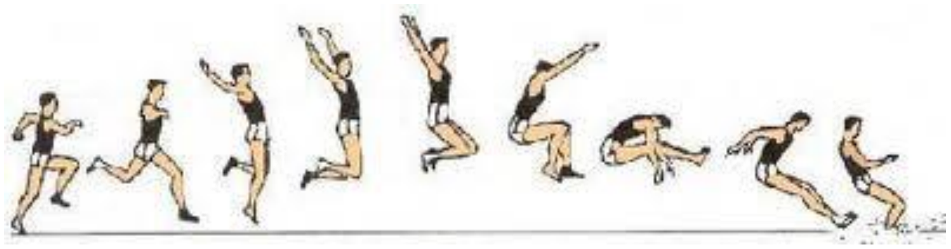
- To store the matrix.
- To perform all mathematical manipulations on matrix.
- Matrix can be multi-dimensional.

# Some classic daily examples…

- 1D
- Running
  - Distance [D]

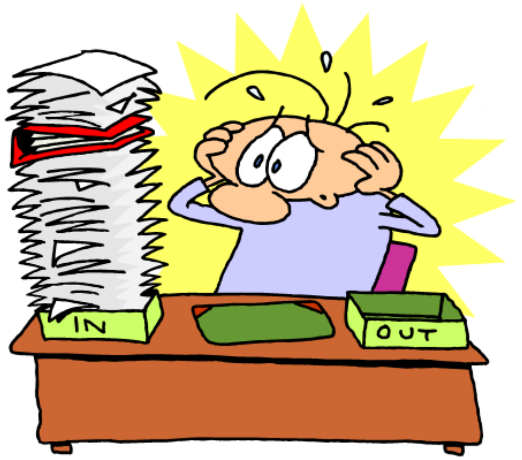


- 2D
- Long Jump
  - Distance, Height [D,H]



- 3D
- High Jump
  - Distance, Height, Roll [D,H,R]

- Next Lecture



Finding my stuffs from a mess …??
Searching Technique

cse101@lpu.co.in