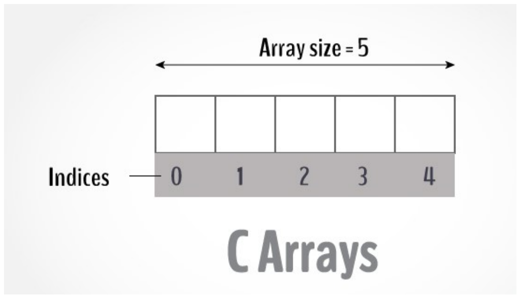
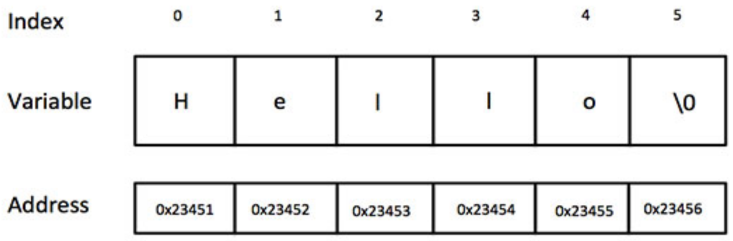
CHAPTER - 4
ARRAY & STRING

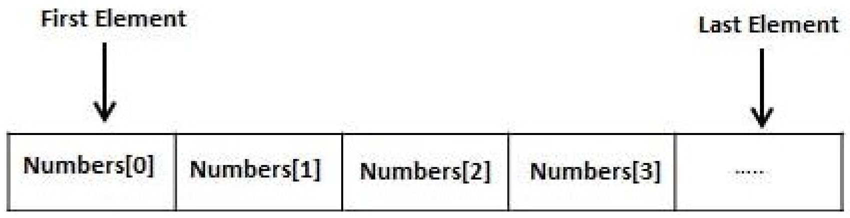




Subject : PPS **Code : 3110003**	**Prepared By:** **Asst. Prof. Rupali Patel** **(CSE Department, ACET)**	AMIRAJ COLLEGE OF ENGINEERING & TECHNOLOGY

Concept of Array

- Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type.
- An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.
- All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.



Concept of Array (cont..)

Declaring Arrays

To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows

Datatype arrayName[arraySize];
Example: double balance[10];

Initializing Arrays

You can initialize an array in C either one by one or using a single statement as follows
double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};

Concept of Array (cont..)

Accessing Array Elements
An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array.

For example :
```
double salary = balance[3];
```

Array Example

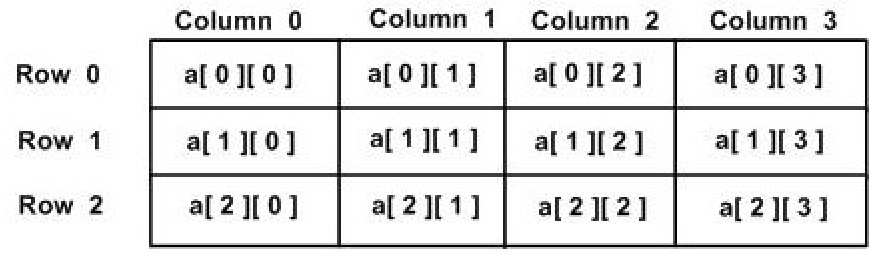
```
#include <stdio.h>
int main()
{
        int values[5];
        printf("Enter 5 integers: ");
        for(int i = 0; i < 5; ++i)
        {
                scanf("%d", &values[i]);
        }
        printf("Displaying integers: ");
        for(int i = 0; i < 5; ++i)
        {
                printf("%d\n", values[i]);
        } return 0;
}
```

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Two-dimensional Arrays

The simplest form of multidimensional array is the two-dimensional array. To declare a two-dimensional integer array of size [x][y], you would write something as follows −
datatype arrayName [3][4];

Where **type** can be any valid C data type and **arrayName** will be a valid C identifier. A two-dimensional array can be considered as a table which will have x number of rows and y number of columns.



	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]

Two-dimensional Arrays (cont..)

Initializing Two-Dimensional Arrays

Multidimensional arrays may be initialized by specifying bracketed values for each row. Following is an array with 3 rows and each row has 4 columns.

```
int a[3][4] = {  {0, 1, 2, 3} ,  {4, 5, 6, 7} ,   {8, 9, 10, 11}   };
```

Accessing Two-Dimensional Array Elements

An element in a two-dimensional array is accessed by using the subscripts, i.e., row index and column index of the array. For example −

```
int val = a[2][3];
```

Two-dimensional Arrays Example

```
#include<stdio.h>
int main()
{
        int i=0,j=0;
        int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};
        for(i=0;i<4;i++)
        {
                for(j=0;j<3;j++)
                {
                        printf("arr[%d] [%d] = %d \n",i,j,arr[i][j]);
                }
        }
        return 0;
}
```

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Strings In C

Strings are defined as an array of characters. The difference between a character array and a string is the string is terminated with a special character ‘\0’.

Declaration of strings:
Declaring a string is as simple as declaring a one dimensional array. Below is the basic syntax for declaring a string
char str_name[size];

Initializing a String:
A string can be initialized in different ways. We will explain this with the help of an example.

```
char c[] = "abcd";
 char c[50] = "abcd";
 char c[] = {'a', 'b', 'c', 'd', '\0'};
 char c[5] = {'a', 'b', 'c', 'd', '\0'};
```

Built-in String Functions

STRLEN():
strlen(s1) calculates the length of string s1.

```
#include <stdio.h>
#include <string.h>
Void main()
{
  char name[ ]= "Hello";
  int len1, len2;
  len1 = strlen(name);
  len2 = strlen("Hello World");
  printf("length of %s = %d\n", name, len1);
  printf("length of %s = %d\n", "Hello World", len2);
  }
```

Built-in String Functions (cont..)

STRCAT():

strcat(s1, s2) concatenates(joins) the second string s2 to the first string s1.

```
#include <stdio.h>
#include <string.h>
int main()
{
        char s2[ ]= "World";
        char s1[20]= "Hello";
        strcat(s1, s2);
        printf("Source string = %s\n", s2);
        printf("Target string = %s\n", s1);
        return 0;
}
```

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Built-in String Functions (cont..)

STRCPY():

strcpy(s1, s2) copies the second string s2 to the first string s1.

```
#include <string.h>
#include <stdio.h>
int main()
{
  char s2[ ]= "Hello";
  char s1[];
  strcpy(s1, s2);
  printf("Source string = %s\n", s2);
  printf("Target string = %s\n", s1);
  return 0;
}
```

Built-in String Functions (cont..)

STRCMP():
strcmp(s1, s2) compares two strings and finds out whether they are same or different.

It compares the two strings character by character till there is a mismatch.

If the two strings are **identical**, it returns a **0**.

If not, then it returns the difference between the ASCII values of the first non-matching pair of characters.

Built-in String Functions (cont..)

Example of strcmp():

```
#include <stdio.h>
#include <string.h>
int main()
{
  char s1[ ]= "Hello";
  char s2[ ]= "World";
  int i, j;
  i = strcmp(s1, "Hello");
  j = strcmp(s1, s2);
  printf("%d \n %d\n", i, j);
  return 0;
}
```

AMIRAJ COLLEGE OF ENGINEERING & TECHNOLOGY

Thank you

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY