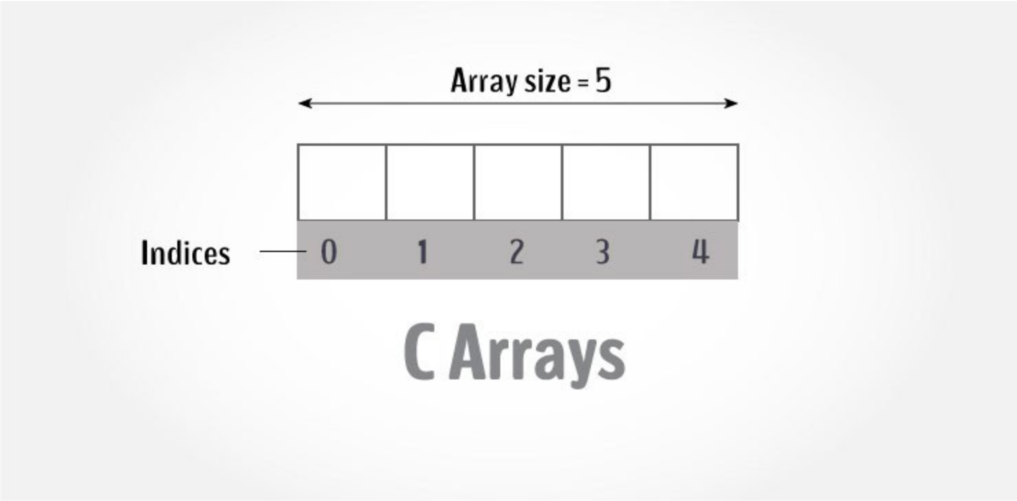
Lecture 3: Arrays

In this tutorial, you will learn to work with arrays. You will learn to declare, initialize and access array elements of an array with the help of examples.

Overview of Array

An array is collection of items sored at continuous memory locations.



Why do we need arrays?

We can use normal variables (v_1, v_2, v_3, ... v_n) when we have a small number of objects, but if we want to sore a large number of instances, it becomes difficult to manage them with normal variables. The idea of an array is to represent many instances in one variable.

The idea behind creating C language was to create an easy language which requires a simple compiler and enables programmers to efficiently interact with the machine/system, just like machine instructions.

C language compiler converts the readable C language program into machine instruction.

How to declare Array in C

```
int num[35];  /* An integer array of 35 elements */
char ch[10];  /* An array of characters for 10 elements */
```

Similarly, an array can be of any data type such as `double`, `float`, `short` etc.

How to access element of an array in C

You can use **array subscript** (or index) to access any element stored in array. Subscript starts with 0, which means arr[0] represents the first element in the array arr. In general, arr[n-1] can be used to access nth element of an array. where n is any integer number.

How to initialize an array?

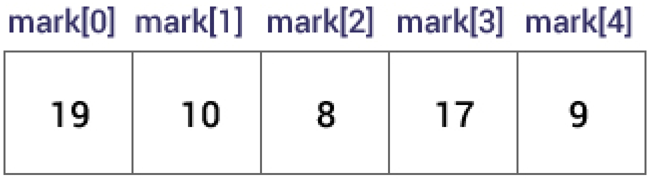
It is possible to initialize an array during declaration. For example,

```
int mark[5] = {19, 10, 8, 17, 9};
```

You can also initialize an array like this.

```
int mark[] = {19, 10, 8, 17, 9};
```

Here, we haven't specified the size. However, the compiler knows its size is 5 as we are initializing it with 5 elements.



mark[0]	mark[1]	mark[2]	mark[3]	mark[4]
19	10	8	17	9

Here,

```
mark[0] is equal to 19
mark[1] is equal to 10
mark[2] is equal to 8
mark[3] is equal to 17
mark[4] is equal to 9
```

Change Value of Array elements

```
int mark[5] = {19, 10, 8, 17, 9}

// make the value of the third element to -1
mark[2] = -1;

// make the value of the fifth element to 0
mark[4] = 0;
```

Input and Output Array Elements

Here's how you can take input from the user and store it in an array element.

```
// take input and store it in the 3rd element
scanf("%d", &mark[2]);

// take input and store it in the ith element
scanf("%d", &mark[i-1]);
```

Here's how you can print an individual element of an array.

```
// print the first element of the array
printf("%d", mark[0]);

// print the third element of the array
printf("%d", mark[2]);

// print ith element of the array
printf("%d", mark[i-1]);
```

Example 1: Array Input/Output

```
// Program to take 5 values from the user and store them in an array
// Print the elements stored in the array
#include <stdio.h>

int main() {
  int values[5];

  printf("Enter 5 integers: ");

  // taking input and storing it in an array
  for(int i = 0; i < 5; ++i) {
     scanf("%d", &values[i]);
  }

  printf("Displaying integers: ");

  // printing elements of an array
  for(int i = 0; i < 5; ++i) {
     printf("%d\n", values[i]);
  }
  return 0;
}
```

Output

```
Enter 5 integers: 1
-3
34
```

```
0
3
Displaying integers: 1
-3
34
0
3
```

Here, we have used a `for` loop to take 5 inputs from the user and store them in an array. Then, using another `for` loop, these elements are displayed on the screen.

Example of Array in C programming to find out the average of 4 integers

```c
#include <stdio.h>
int main()
{
    int avg = 0;
    int sum =0;
    int x=0;

    /* Array- declaration – length 4*/
    int num[4];

    /* We are using a for loop to traverse through the array
     * while storing the entered values in the array
     */
    for (x=0; x<4;x++)
    {
        printf("Enter number %d \n", (x+1));
        scanf("%d", &num[x]);
    }
    for (x=0; x<4;x++)
    {
        sum = sum+num[x];
    }

    avg = sum/4;
    printf("Average of entered number is: %d", avg);
    return 0;
}
```

Output:

```
Enter number 1
10
Enter number 2
10
Enter number 3
20
Enter number 4
40
```

```
Average of entered number is: 20
```

Let’s discuss the important parts of the above program:

Input data into the array

Here we are **iterating the array** from 0 to 3 because the size of the array is 4. Inside the loop we are displaying a message to the user to enter the values. All the input values are stored in the corresponding array elements using scanf function.

```
for (x=0; x<4;x++)
{
    printf("Enter number %d \n", (x+1));
    scanf("%d", &num[x]);
}
```

Reading out data from an array

Suppose, if we want to display the elements of the array then we can use the for loop in C like this.

```
for (x=0; x<4;x++)
{
    printf("num[%d]\n", num[x]);
}
```

Various ways to initialize an array

In the above example, we have just declared the array and later we initialized it with the values input by user. However, you can also initialize the array during declaration like this:

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

OR (both are same)

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

Un-initialized array always contains garbage values.