CL1002 – Programming Fundamentals Lab

Lab # 10

Arrays

Instructor: Engr. Muhammad Usman

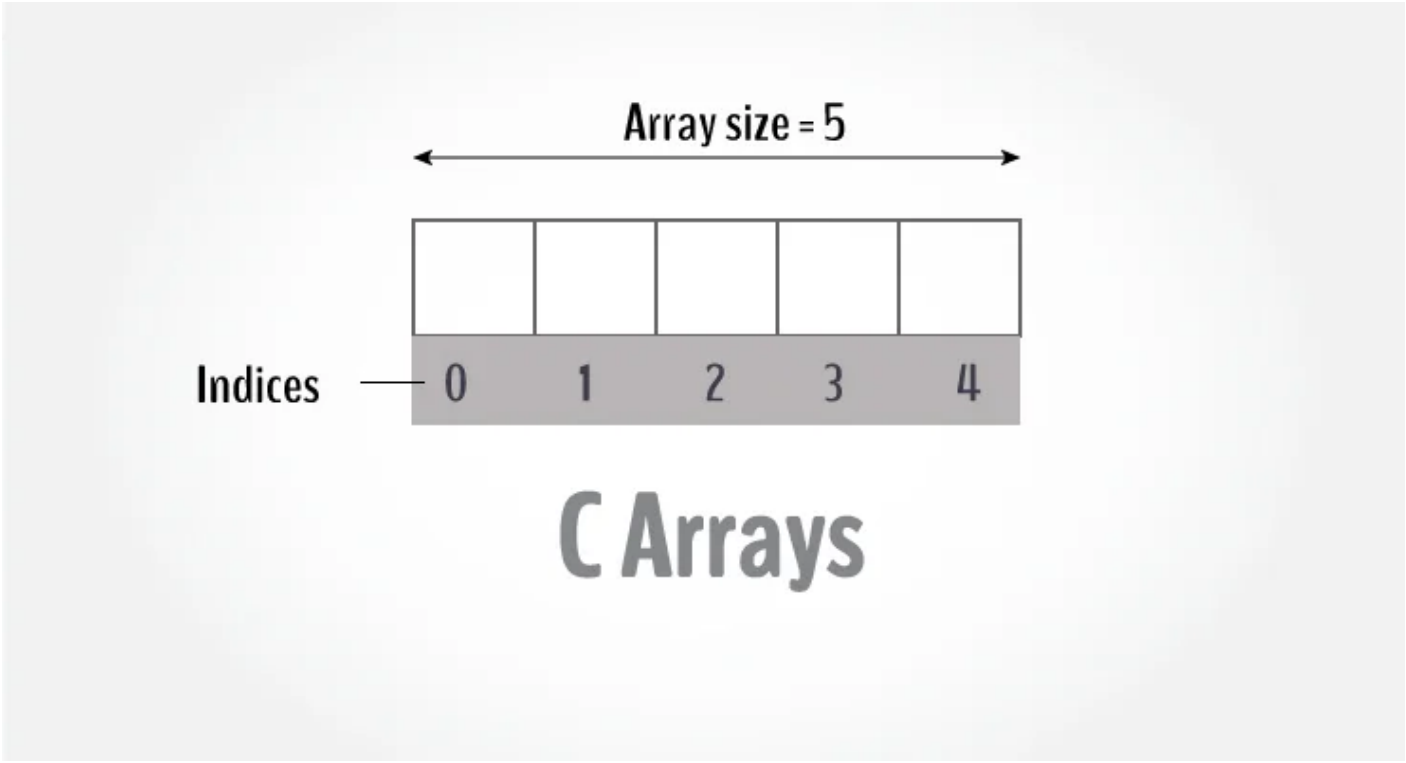
Email: usman.rafiq@nu.edu.pk

Department of Computer Science,

National University of Computer and Emerging Sciences FAST Peshawar

Arrays in C/C++

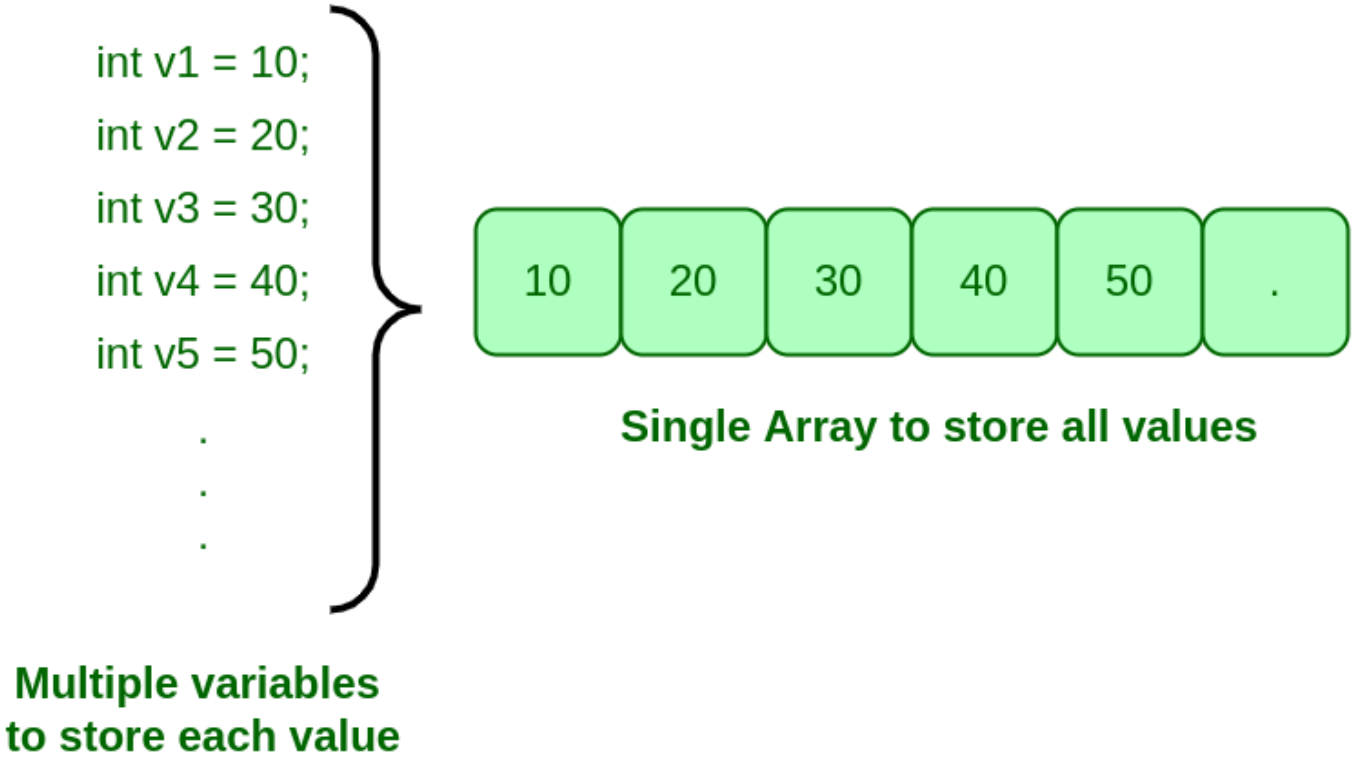
1. It is a group of variables of similar data types referred to by a single element.
2. Its elements are stored in a contiguous memory location.
3. The size of the array should be mentioned while declaring it.
4. Array elements are always counted from zero (0) onward.
5. Array elements can be accessed using the position of the element in the array.
6. The array can have one or more dimensions.



An array in C/C++ or be it in any programming language is a collection of similar data items stored at contiguous memory locations and elements can be accessed randomly using indices of an array.

Why do we need arrays?

We can use normal variables (v1, v2, v3, ..) when we have a small number of objects, but if we want to store 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.



Advantages:-

- Code Optimization: we can retrieve or sort the data efficiently.
- Random access: We can get any data located at an index position.

Disadvantages:-

- Size Limit: We can store only the fixed size of elements in the array. It doesn't grow its size at runtime.

An array is a variable that can store multiple values. For example, if you want to store 100 integers, you can create an array for it.

```
int data[100];
```

How to declare an array?

```
dataType arrayName[arraySize];
```

For example,

```
float mark[5];
```

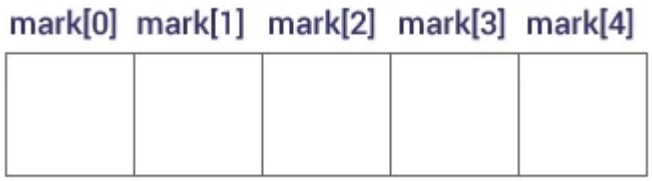
Here, we declared an array, mark, of floating-point type. And its size is 5. Meaning, it can hold 5 floating-point values.

It's important to note that the size and type of an array cannot be changed once it is declared.

Access Array Elements

You can access elements of an array by indices.

Suppose you declared an array mark as above. The first element is mark[0], the second element is mark[1] and so on.



Few keynotes:

- Arrays have 0 as the first index, not 1. In this example, **mark[0]** is the first element.
- If the size of an array is **n**, to access the last element, the **n-1** index is used. In this example, **mark[4]**

How to initialize an array?

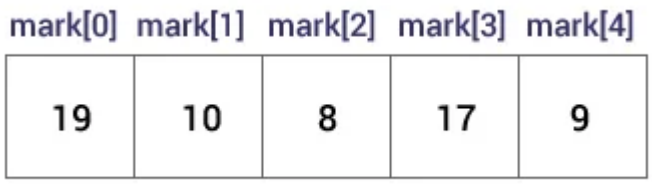
It is possible to initialize an array during declaration. For example,

```
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.



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;
```

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:\n");

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

Output

```
Enter 5 integers: 7 2 9 1 8
Displaying integers:
7  2  9  1  8
```

Example 2 | Calculate Sum

```c
// Program to find the sum of 5 numbers using arrays
#include <stdio.h>
int main() {
 int marks[5], i, n, sum = 0;
  for(i=0; i < 5; ++i) {
   printf("Enter number%d: ",i+1);
   scanf("%d", &marks[i]);

   // adding integers entered by the user to the sum variable
   sum += marks[i];
 }
 printf("Sum = %d", sum);
 return 0;
}
```

Output

```
Enter number1: 3
Enter number2: 5
Enter number3: 1
Enter number4: 2
Enter number5: 8
Sum = 19
```

References:

https://www.programiz.com/c-programming/c-arrays