C++ Arrays

In this tutorial, we will learn to work with arrays. We will learn to declare, initialize, and access array elements in C++ programming with the help of examples.

In C++, an array is a variable that can store multiple values of the same type. For example,

Suppose a class has 27 students, and we need to store the grades of all of them. Instead of creating 27 separate variables, we can simply create an array:

```
double grade[27];
```

Here, grade is an array that can hold a maximum of 27 elements of double type.

In C++, the size and type of arrays cannot be changed after its declaration.

C++ Array Declaration

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

For example,

```
int x[6];
```

Here,

int - type of element to be stored

x - name of the array

6 - size of the array

Few Things to Remember:

- The array indices start with 0. Meaning x[0] is the first element stored at index 0.
- If the size of an array is n, the last element is stored at index (n-1). In this example, x[5] is the last element.
- Elements of an array have consecutive addresses. For example, suppose the starting address of x[0] is 2120.

- Then, the address of the next element x[1] will be 2124, the address of x[2] will be 2128, and so on.
- Here, the size of each element is increased by 4. This is because the size of int is 4 bytes.

C++ Array Initialization

In C++, it's possible to initialize an array during declaration. For example,

```
// declare and initialize and array
int x[6] = {19, 10, 8, 17, 9, 15};
```

Another method to initialize array during declaration:

```
// declare and initialize an array
int x[] = {19, 10, 8, 17, 9, 15};
```