Creating an Array

In this chapter, you will see the implementation of arrays in C++.

We'll cover the following

Introduction#

An **array** is a collection of elements of the same data type a the single name. Let's see how we can declare and initialize an array in C++.

Array declaration#

The general syntax for declaring an array is given below:

```
DataType  ArrayName  [ ArraySize ] ;
```

In the array declaration, we specify the data type followed by an array name, which is then followed by an array size in square brackets.

See the code given below!

```
#include <iostream>

using namespace std;

int main() {

  int Roll_Number[5];

}
```

We declare an array `Roll_Number` that can store **5** integer values. The compile reserves space for **5** elements of type `int` consecutively in memory. Since the data type of an element is `int`, it reserves **4 bytes** for each element, and in total, it reserves **5*4** = **20 bytes** with the name `Roll_Number`. Since an array can store **5** elements, the size of an array is **5**.

Array initialization#

Approach 1

We can assign a value to an array element by accessing its index.

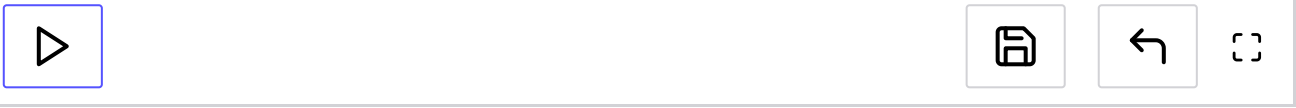
ArrayName [ArrayIndex] = value ;

See the code given below!

```
#include <iostream>

```

```
 3  using namespace std;
 4
 5  int main() {
 6
 7    int Roll_Number[5];
 8
 9    Roll_Number[0] = 100;
10    Roll_Number[1] = 101;
11    Roll_Number[2] = 102;
12    Roll_Number[3] = 103;
13    Roll_Number[4] = 104;
14
15  }
```



The code above initializes an array `Roll_Number` that stores:

100 at index `0`

101 at index `1`

102 at index `2`

103 at index `3`

104 at index `4`

Approach 2

You must be wondering if we can just declare and initialize all elements in an array in one go. The answer is yes.

We can assign a value to the array elements in the declaration step.

```
DataType  ArrayName  [ ] = { value1, value2, ..... valueN } ;
```

In the code given below, we will initialize an array `Roll_Number` in the declaration step.

```
#include <iostream>

using namespace std;

int main() {

  int Roll_Number[ ] = { 100, 101, 102, 103, 104 };

}
```

ℹ️ If we are initializing an array in the declaration step, we don't need to specify the size of the array. The compiler automatically determines its size.

Initializing an array with fewer elements than its total size#

If we initialize an array with elements fewer than its total size, the compiler automatically initializes the remaining elements with their default values.

```
#include <iostream>

using namespace std;

int main() {

  int Roll_Number[5] = {100, 101};

```

```
9  }
```

In the code above, even though we have not initialized the values from index **2 to 4**, the compiler automatically initializes them to their default values.

> 📝 If we specify the size of an array in the declaration step and then initialize more values than the specified size, the compiler will generate an error.

Quiz

(/learn)

Q What does the following statement do?

```
int main() {
  int Roll_Number[5] = { 100, 101, 102, 103, 104, 105 };

}
```

A) Creates an array of **5** values `{100, 101, 102, 103, 104}`

B) Creates an array of **6** values `{100, 101, 102, 103, 104, 105}`

Your Answer

C) Generates an error

Explanation

If we specify the size of an array in the declaration step and then initialize more values than the specified size compiler will generate an error.

In the code above, the size of an array is `5`, and we have specified `6` values in the curly braces.

Submit Answer

Reset Quiz

That is all about creating an array in C++. In the next lesson, you will learn how to access and update elements stored in an array.

Back

Introduction to Arrays

Next

Accessing an Array

Mark as Completed

Report an Issue