Accessing an Array

In this lesson, you will learn how to access and update the elements stored in an array.

We'll cover the following

Array traversal

We can access the elements stored in an array by writing the array name, which is followed by its index in the square brackets.

```
ArrayName [ Index ] ;
```

The first element of an array is stored at index **0**, and the last element is stored at index **size-1**. **Array[0]** refers to the first element in an array. **Array[1]** refers to the second element, and so on.

Example program

Suppose there are **5** elements in an array, then the index value of this array will range from **0 to 4**.

```
#include <iostream>

using namespace std;

int main(){

  int Roll_Number[5] = {100, 101, 102, 103, 104};

  cout << Roll_Number[4];
}
```

The code given above accesses the element at an index of four, which is the 5^{th}

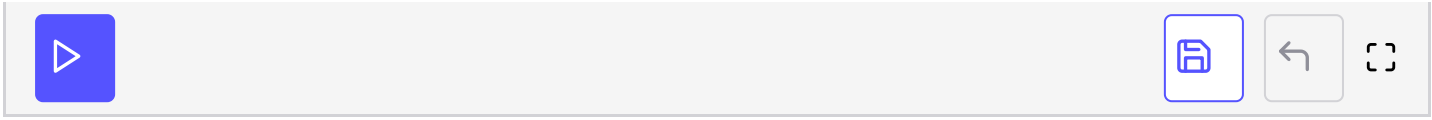
element of an array.

> If you try to access an index that is greater than the size of an array, then the compiler will not generate an error, but it may give you an unexpected output.

Print all elements of an array using for loop

Accessing each and every element in an array and then printing its value is a repetitive task. So, let's write a code that does print all the elements of an array using the `for` loop.

```
#include <iostream>

using namespace std;

int main() {

  int size = 5;
  //Initialize array
  int Roll_Number[] = {100, 101, 102, 103, 104};

  //Print Array
  for (int i = 0; i < size; i++) {
    // Access element at index i
    cout << Roll_Number[i] << " ";
  }
  cout << endl;
}
```



Line No. 9: Initializes an array `Roll_Number` that can store **5** values.

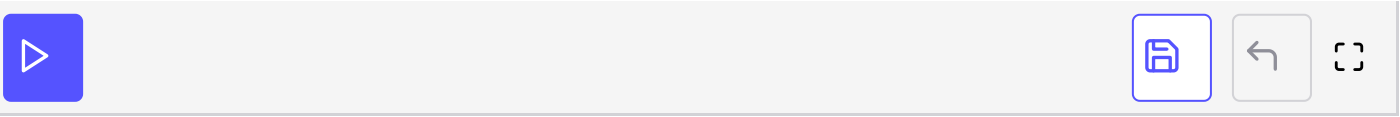
Line No. 12: The `for` loop iterates from **i** = **0** to **i** = **size-1**. In the loop body, we are printing the array `Roll_Number` element at index `i`.

Array updation

We can change the value of the elements in an array by accessing its index and assigning a new value to that array index.

```
#include <iostream>

using namespace std;
```

```
int main() {

  int size = 5;

  // Initialize array
  int Roll_Number[size] = {100, 101, 102, 103, 104};

  cout << "Values of array before updation: " << endl;
  // Print values of array
  for (int i = 0; i < size; i++) {
    // Accesss elements of array at index i
    cout << Roll_Number[i] << "  ";
  }
  cout << endl;
  // Update values of array element at index 3 and 4
  Roll_Number[3] = 22222;
  Roll_Number[4] = 33333;
  cout << "Values of array after updation: " << endl;
  // Print updated values of  array
  for (int i = 0; i < size; i++) {
    // Access elements of array at index i
    cout << Roll_Number[i] << "  ";
  }
  cout << endl;
}
```



In the code above:

Line No. 9: Initializes array `Roll_Number` that can store **5** elements

Line No. 13: The `for` loop prints the elements of an array.

Line No. 19: Updates the value of array `Roll_Number` at index `3` to **22222**

Line No. 20: Updates the value of array `Roll_Number` at index `4` to **33333**

Line No. 23: The `for` loop prints the updated values of an array.

Quiz

What is the output of the following code?

```
{
  int size = 5;
  int number[size] = {1,2,3,4,5};
  number[3] = -6;
```

```
  for (int i = 0; i < size; i++) {
    cout << number[i] << "  ";

  }
}
```

Retake Quiz

Interesting so far? Read on to the next lesson to see how to pass an array to a function.