C++ Introduction
C++ Flow Control
C++ Functions
- C++ Functions
- C++ Function Types
- C++ Function Overloading
- C++ Default Argument
- C++ Storage Class
- C++ Recursion
- C++ Return Reference

C++ Arrays & String
C++ Structures
C++ Object & Class
C++ Pointers
C++ Inheritance

Related Topics
- C++ Storage Class
- C++ Call by Reference: Using pointers
- C++ Structure and Function
- How to pass and return object from C++ Functions?
- C++ Function Overloading
- Swap Numbers in Cyclic Order Using Call by Reference

C++ Return by Reference

In this article, you'll learn how to return a value by reference in a function and use it efficiently in your program.

In C++ Programming, not only can you pass values by reference to a function but you can also return a value by reference.

To understand this feature, you should have the knowledge of:

- Global variables

Example: Return by Reference

```
#include <iostream>
using namespace std;

// global variable
int num;

// function declaration
int& test();

int main() {

  // assign 5 to num variable
  // equivalent to num = 5;
  test() = 5;

  cout << num;

  return 0;
}

// function definition
// returns the address of num variable
int& test() {
  return num;
}
```

Output

```
5
```

In program above, the return type of function `test()` is `int&`. Hence, this function returns a reference of the variable `num`.

The return statement is `return num;`. Unlike return by value, this statement doesn't return value of `num`, instead it returns the variable itself (address).

So, when the **variable** is returned, it can be assigned a value as done in `test() = 5;`

This stores **5** to the variable `num`, which is displayed onto the screen.

Important Things to Remember When Returning by Reference.

- Ordinary function returns value but this function doesn't. Hence, you cannot return a constant from the function.

```
int& test() {
    return 2;
}
```

- You cannot return a local variable from this function.

```
int& test() {
    int n = 2;
    return n;
}
```

Previous Tutorial: C++ Recursion

Next Tutorial: C++ Arrays

Related Tutorials

C++ Tutorial
C++ Storage Class

C++ Tutorial
C++ Call by Reference: Using pointers

C++ Tutorial
C++ Structure and Function

C++ Tutorial
C++ Multidimensional Arrays