

RETURNING VALUES BY REFERENCE IN C++

http://www.tutorialspoint.com/cplusplus/returning_values_by_reference.htm

Copyright © tutorialspoint.com

A C++ program can be made easier to read and maintain by using references rather than pointers. A C++ function can return a reference in a similar way as it returns a pointer.

When a function returns a reference, it returns an implicit pointer to its return value. This way, a function can be used on the left side of an assignment statement. For example, consider this simple program:

```
#include <iostream>
#include <ctime>

using namespace std;

double vals[] = {10.1, 12.6, 33.1, 24.1, 50.0};

double& setValues( int i )
{
    return vals[i];    // return a reference to the ith element
}

// main function to call above defined function.
int main ()
{
    cout << "Value before change" << endl;
    for ( int i = 0; i < 5; i++ )
    {
        cout << "vals[" << i << "] = ";
        cout << vals[i] << endl;
    }

    setValues(1) = 20.23; // change 2nd element
    setValues(3) = 70.8;  // change 4th element

    cout << "Value after change" << endl;
    for ( int i = 0; i < 5; i++ )
    {
        cout << "vals[" << i << "] = ";
        cout << vals[i] << endl;
    }
    return 0;
}
```

When the above code is compiled together and executed, it produces the following result:

```
Value before change
vals[0] = 10.1
vals[1] = 12.6
vals[2] = 33.1
vals[3] = 24.1
vals[4] = 50
Value after change
vals[0] = 10.1
vals[1] = 20.23
vals[2] = 33.1
vals[3] = 70.8
vals[4] = 50
```

When returning a reference, be careful that the object being referred to does not go out of scope. So it is not legal to return a reference to local var. But you can always return a reference on a static variable.

```
int& func() {  
    int q;  
    //! return q; // Compile time error  
    static int x;  
    return x;      // Safe, x lives outside this scope  
}
```