



## Pointers in C++ | C++ Tutorials for Beginners #12



[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

## Pointers in C++ | C++ Tutorials for Beginners #12

In this series of our C++ tutorials, we will visualize pointers in the C++ language in this lecture. In our last lesson, we discussed break statements and continue statements in C++.

### Pointers in C++

A pointer is a data type which holds the address of other data type. The “&” operator is called “**address of**” operator, and the “\*” operator is called “**value at**” dereference operator. An example program for pointers is shown in figure 1.

```
int a=3;  
int* b = &a;
```

```
cout<<b;
```

**Figure 1: Pointer Program**

As shown in figure 1, at 1<sup>st</sup> line an integer variable “**a**” is initialized with the value “**3**”. At the 2<sup>nd</sup> line, the address of integer variable “**a**” is assigned to the integer pointer variable “**b**”. At the 3<sup>rd</sup> line, the address of the integer pointer variable “**b**” is printed. The output of the following program is shown in figure 2.

```
PS D:\Business\code pla  
0x61ff08  
PS D:\Business\code pla
```

**Figure 2: Pointer Program Output**

As shown in figure 2, the address of the integer pointer variable “**b**” is printed. The main thing to note here is that the address printed by the variable “**b**” is the address of integer variable “**a**” because we had assigned the address of variable “**a**” to the integer pointer variable “**b**”. To clarify, we will print both variable “**a**” and variable “**b**” addresses, which are shown in figure 3.

```
int a=3;  
int* b = &a;  
cout<<"The address of a is "<<&a<<endl;  
cout<<"The address of a is "<<b<<endl;
```

**Figure 3: Pointer Program Example 2**

As shown in figure 3, now we printed both variable “**a**” and variable “**b**” addresses. The output for the following program is shown in figure 4.

```
PS D:\Business\code playground  
The address of a is 0x61ff08  
The address of a is 0x61ff08
```

**Figure 4: Pointer Program Example 2 Output**

As shown in figure 4, both variables “**a**” and “**b**” have the same addresses, but in actual, this is the address of the variable “**a**”, the variable “**b**” is just pointing to the address of the variable “**a**”.

To see the value of variable “**a**” using a pointer variable, we can use the “**\***” dereference operator. An example of

the dereference operator program is shown in figure 5.

```
// * ---> (value at) Dereference operator
cout<<"The value at address b is "<<*b<<endl;
```

**Figure 5: Dereference Operator example**

As shown in figure 5, the value at address “b” is printed. The main thing to note here is that the value printed by the pointer variable “b” will be the value of variable “a” because the pointer variable “b” is pointing to the address of the variable “a”. The output for the following program is shown in figure 6.

```
// * ---> (value at) Dereference operator
cout<<"The value at address b is "<<*b<<endl;
```

**Figure 6: Dereference Operator Example**

## Pointer to Pointer

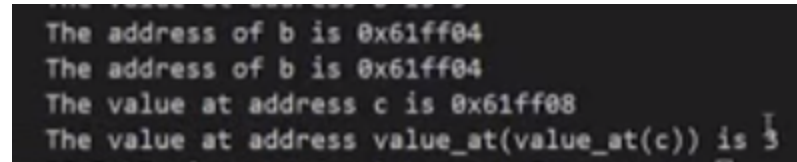
Pointer to Pointer is a simple concept, in which we store the address of one Pointer to another pointer. An example program for Pointer to Pointer is shown in figure 7.

```
// Pointer to pointer
int** c = &b;
cout<<"The address of b is "<<&b<<endl;
cout<<"The address of b is "<<c<<endl;
cout<<"The value at address c is "<<*c<<endl;
cout<<"The value at address value_at(value_at(c)) is "<<**c<<endl;
```

**Figure 7: Pointer to Pointer Example Program**

As shown in figure 7, at the 1<sup>st</sup> line, the address of the pointer variable “b” is assigned to the pointer variable “c”. At 2<sup>nd</sup> line, the address of the pointer variable “b” is printed. At the 3<sup>rd</sup> line, the address of the pointer variable “c” is printed. At line 4<sup>th</sup>, the value at the pointer variable “c” is printed. At line 5<sup>th</sup>, the pointer variable “c” will be

dereferenced two times, and it will print the value at pointer variable "**b**". The output of the following program is shown in figure 2. The output for the following program is shown in figure 8.



```
The address of b is 0x61ff04
The address of b is 0x61ff04
The value at address c is 0x61ff08
The value at address value_at(value_at(c)) is 3
```

**Figure 8: Pointer to Pointer Example Program Output**

**Code as described/written in the video**

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

int main(){
    // What is a pointer? ----> Data type which holds the address of other data types
    int a=3;
    int* b;
    b = &a;

    // & ----> (Address of) Operator
    cout<<"The address of a is "<<&a<<endl;
    cout<<"The address of a is "<<b<<endl;

    // * ----> (value at) Dereference operator
    cout<<"The value at address b is "<<*b<<endl;

    // Pointer to pointer
```

```
int** c = &b;
cout<<"The address of b is "<<&b<<endl;
cout<<"The address of b is "<<c<<endl;
cout<<"The value at address c is "<<*c<<endl;
cout<<"The value at address value_at(value_at(c)) is "<<**c<<endl;

return 0;
}
```

[Previous](#)[Next](#)

CodeWithHarry

Copyright © 2022 CodeWithHarry.com

