

Cursor NULL in C / C ++

It is always a good practice to assign a NULL pointer to a pointer variable in case you don't know the exact address to be assigned. This is done at the time of variable declaration. A pointer that is assigned NULL is called a null pointer.

It is always a good practice to assign a NULL pointer to a pointer variable in case you don't know the exact address to be assigned. This is done at the time of variable declaration. A pointer that is assigned NULL is called a **null pointer** .

The NULL pointer is a constant with a value of 0 defined in several standard libraries, including **iostream** . You consider the following example:

```
#include using namespace std ; int main () { int * contro = NULL ; cout
```

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

```
Gia tri cua contro la 0
```

On most operating systems, programs are not allowed to access memory at address 0, because that memory is reserved by the operating system. However, memory address 0 has special meaning; it signals that the pointer cannot point to an accessible memory location. But by convention, if a pointer contains a value of 0, it is treated as not pointing to anything.

To test a null pointer, you can use an if statement as follows:

```
if ( contro ) // true neu contro khong la NULL if (! contro ) // true neu contr
```

If all unused cursors are provided with null values ??and you avoid using null pointers, you can avoid the misuse of an uninitialized pointer. Many times, uninitialized variables hold some junk values, and it makes it more difficult to debug the program.

According to Tutorialspoint

Previous article: [Cursor in C / C ++](#)

Next lesson: [Arithmetic pointer in C / C ++](#)

You finished reading the article "**Cursor NULL in C / C ++**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.