

The "null" Pointer

The value that indicates that a pointer is not being used is called the **null pointer**. It is represented internally by `0`. While you **cannot assign an arbitrary integer** to a pointer variable, you **can assign the value 0**.

Using a literal `0`, however, makes it hard to find all of the null pointers in your code. C++11 introduced **an actual null pointer constant** named `nullptr`. You should use that instead `0`. Do not use the C language value `NULL`.

It is **illegal to dereference** a null pointer. In UNIX, it usually results in a **segmentation fault**, but that is **not guaranteed**. Some machines return the contents of address `0000`. As a result, **this is undefined behavior**, as in the case of uninitialized pointers.

If you declare a pointer but **fail to initialize it**, the computer tries to interpret the contents of that pointer as an address and tries to read that region of memory. Such programs can fail in ways that are extremely difficult to detect. Again, this is **undefined behavior**.



This course content is offered under a [CC Attribution Non-Commercial](#) license. Content in this course can be considered under this license unless otherwise noted.