# Pointers and Const-Correctness

Pointers have two modes of const-ness: pointers that do not allow modifications to the data, and pointers that must always point to the same address. The two may be combined. For the full story on const-correctness, see const correctness--why bother?

## Pointer to Constant Data

A pointer to const data does not allow modification of the data through the pointer. The declaration of const data merely requires that the const precede the *, so either of the following two declarations are valid.

```
const type* variable;
```

or

```
type const * variable;
```

The memory address stored in a pointer to constant data cannot be assigned into regular pointers (that is, pointers to non-const data) without a const cast.

## Pointers with Const Memory Address

Pointers with a constant memory address are declared by including the const after the *. Because the address is const, the pointer must be assigned a value immediately.

```
type * const variable = some memory address;
```

## Const Data with a Const Pointer

To combine the two modes of const-ness with pointers, you can simply include const for both data and pointer by putting const both before and after the *:

```
const type * const variable = some memory address;
```

or

```
type const * const variable = some memory address;
```

**Popular pages**

- Jumping into C++, the Cprogramming.com ebook
- How to learn C++ or C
- C Tutorial
- C++ Tutorial
- 5 ways you can learn to program faster
- The 5 most common problems new programmers face
- How to set up a compiler
- How to make a game in 48 hours