

Re: Interesting question on const.

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-03/2699.html

From: Tor Rustad (*torust_at_online.no.spam*)

Date: 03/23/05

Date: Wed, 23 Mar 2005 11:42:31 +0100

<vashwath@rediffmail.com> wrote in message

> *Hi all,*
> *Recently I attended an interview in which the question*
> *"Is there any difference between "const T var" and*
> *"T const var"?"*

The answer is that it depends.

Let say T is a typedef for "int", then we have

```
const int var; /* declare read-only int */  
int const var; /* declare read-only int */
```

which is exactly the same thing. "const" type-qualify it's left, but the special case of "const" being first in the declaration, it type-qualify it's right.

Now let say T is a typedef for "int *". Then we have these two cases:

```
int * const var; /* declare read-only int pointer */  
const int *var; /* declare pointer to read-only int */
```

In the first case, the int pointer is const, while in the latter case, int is const. This isn't the same thing!

The const pointer can be initialized to point to an object (which may be non-const), but cannot safely be changed to point to another object later. OTOH, the pointer to a const int object, can safely be changed to point to other const int objects.

--

Tor <torust AT online DOT no>