

HOME

ARTICLES

FORUMS

COMMUNITY

MEMBERS

NEWSLETTER

Operating System

Database

Ethical hacking Tips

Game Programming

Engineering Concepts

Internet Marketing

Home

Articles

Programming

C

This site uses cookies. By continuing to use this site, you are agreeing to our use of cookies. [Learn More.](#)

## Difference Between Constant Pointers and Pointer to Constants

Tags: [constant pointer](#) [constant pointers](#) [pointer to constant](#) [pointer to constants](#)

**poornaMoksha**

New Member

JOINED: Jan 29, 2011

MESSAGES: 150

LIKES RECEIVED: 33

TROPHY POINTS: 0

OCCUPATION:

Software developer

LOCATION: India

This article assumes that readers are aware of what pointers are and how they are used in 'C'. As a quick overview, A pointer is a special variable which holds the address of a variable of same type. For example :

Lets say there is a variable 'int a=10' so,a pointer to variable 'a' can be defined as 'int\* ptr = &a'.

- '&' operator gives you the address of the variable on which this operator is applied
- '\*' operator is the value-of operator and fetches the value kept at address held by the variable it is applied on.

For example, in the above example, \*ptr will give the value of 'a' ie 10.

Now, coming back to the topic. This articles tries to quickly explain the difference between two types of pointers :

- Constant pointers
- Pointer to Constants

Let us take it one by one.

**1) Constant Pointers :** These type of pointers are the one which cannot change address they are pointing to. This means that suppose there is a pointer which points to a variable (or stores the address of that variable). Now

the pointer 'ptr' as constant pointer,

**2) Pointer to Constant :** These type of pointers are the one which cannot change the value they are pointing to. This means they cannot change the value of the variable whose address they are holding.

A pointer to a constant is declared as : 'const int \*ptr' (the location of 'const' makes the pointer 'ptr' as a pointer to constant).

## Examples

Lets look at some quick examples to understand the above two concepts more clearly :

### 1) Constant pointers :

Code:

```
#include<stdio.h>

int main(void)
{
    int a[] = {10,11};
    int* const ptr = a;

    *ptr = 11;

    printf("\n value at ptr is : [%d]\n",*ptr);
    printf("\n Address pointed by ptr : [%p]\n",(unsigned int*)ptr);

    ptr++;
    printf("\n Address pointed by ptr : [%p]\n",(unsigned int*)ptr);

    return 0;
}
```

Now, when we compile the above code, compiler complains :

```
practice # gcc -Wall constant_pointer.c -o constant_pointer
```

```
constant_pointer.c: In function 'main':
```

```
constant_pointer.c:13: error: increment of read-only variable 'ptr'
```

Hence we see very clearly above that compiler complains that we cannot changes the address held by a constant pointer.

```
int main(void)
{
    int a = 10;
    const int* ptr = &a;

    printf("\n value at ptr is : [%d]\n",*ptr);
    printf("\n Address pointed by ptr : [%p]\n",(unsigned int*)ptr);

    *ptr = 11;

    return 0;
}
```

Now, when the above code is compiled, the compiler complains :

```
practice # gcc -Wall pointer_to_constant.c -o pointer_to_constant
```

```
pointer_to_constant.c: In function 'main':
```

```
pointer_to_constant.c:12: error: assignment of read-only location '*ptr'
```

Hence here too we see that compiler does not allow the pointer to a constant to change the value of the variable being pointed.

## A little hack

Have you guyz heard about constant variables? Most of you might have, but for those who haven't :

constant variables are the one whose values cannot be changed. For example:

```
const int a = 10;
```

The above line declares a constant variable 'a' with value 10. Now, since its a constant variable so any attempt to change the value of a will be disregarded by compiler. Lets try :

Code:

```
#include<stdio.h>

int main(void)
{
    const int a = 10;
    a++;
}
```

practice # gcc -Wall hack\_ptr.c -o hack\_ptr

hack\_ptr.c: In function 'main':

hack\_ptr.c:2: error: increment of read-only variable 'a'

Get It now

We see that the compiler clearly says that we cannot change value of a read-only(const) variable 'a'.  
Try all new Adobe Acrobat DC desktop free for 30 days

Now, the interesting part. Can we have a mechanism through which we can change the value of 'a' without the compiler complaining???

Hmmmmm...lets try pointers. Have a look at the following code :

Code:

```
#include<stdio.h>

int main(void)
{
    const int a = 10;
    int* ptr = &a;

    printf("\n value at ptr is  : [%d]\n",*ptr);
    printf("\n Address pointed by ptr  : [%p]\n",(unsigned int*)ptr);

    *ptr = 11;
    printf("\n value at ptr is  : [%d]\n",*ptr);

    return 0;
}
```

Now, this is left as an exercise to the readers to see if the above program actually works?? If yes, then whats the reason??

## Conclusion

To conclude, pointers are a powerful mechanism in 'C' to manipulate memory locations and values present at these locations. 'C' provides the concept of constant pointers and pointer to constants to have control on how pointers are being manipulated.

Any one with questions or suggestions is most welcome!!!!

**jhon786**

New Member

JOINED: Oct 12, 2011  
MESSAGES: 46  
LIKES RECEIVED: 0  
TROPHY POINTS: 0

Thanks for telling the useful information and difference between Constant Pointers and Pointer to Constants.

jhon786, Oct 13, 2011

SHARE #2

**bernaberna1**

New Member

JOINED: Oct 20, 2011  
MESSAGES: 1  
LIKES RECEIVED: 0  
TROPHY POINTS: 0

It's still sounds confusing but i'll get into it more in details. Thanks for posting.

bernaberna1, Oct 20, 2011

SHARE #3

**poornaMoksha**

New Member

JOINED: Jan 29, 2011  
MESSAGES: 150  
LIKES RECEIVED: 33  
TROPHY POINTS: 0  
OCCUPATION: Software developer  
LOCATION: India

jhon786 said: ↑

*Thanks for telling this useful information and difference between Constant Pointers and Pointer to Constants.*

You are welcome buddy!!!!

poornaMoksha, Nov 23, 2011

SHARE #4

**hakz\_exp**

New Member

JOINED: Feb 17, 2012

MESSAGES: 1

LIKES RECEIVED: 0

TROPHY POINTS: 0

hakz\_exp, Feb 17, 2012

SHARE #5

**Scripting**

John Hoder

JOINED: Jun 29, 2010

MESSAGES: 421

LIKES RECEIVED: 57

TROPHY POINTS: 0

OCCUPATION:

School for life

LOCATION: /root

Remarkable tutorial, and yes I think too that the hack won't work

Scripting, Feb 18, 2012

SHARE #6

**lionaneesh**

Active Member

JOINED: Mar 21, 2010

MESSAGES: 848

LIKES RECEIVED: 224

TROPHY POINTS: 43

OCCUPATION: Student

LOCATION: India

I just tried and it works, but I can't quite think of a reason. :-?

lionaneesh, Jun 25, 2012

SHARE #7

Sign up now!



## MEMBERS ONLINE NOW

Total: 295 (members: 0, guests: 249, robots: 46)

## NEW PROFILE POSTS



**AnjaliK** I am a engineer I would like travel and while traveling I read and write on various topics. It is very nice to have community of good people  
Sep 12, 2017

...



**mickeyy** Hi!  
Sep 9, 2017

...



**Arihant Tripathi** Omega Softwares is Mumbai based Software Development Company, and also provide Website designing and development, ERP, CRM, MLM Software  
Aug 17, 2017

...



**RED monkey** wanna!!! haCk...but not possible.....  
Aug 6, 2017

...



**ncodetechnologiesinc** NCode Technologies is renowned Web Development Company based in India, providing Offshore Web Designs Development  
Jul 19, 2017

...

## FORUM STATISTICS

DISCUSSIONS:	24,726
MESSAGES:	96,397





