

# Const Qualifier in C

Difficulty Level : Medium • Last Updated : 30 Jul, 2021

The qualifier const can be applied to the declaration of any variable to specify that its value will not be changed ( Which depends upon where const variables are stored, we may change the value of const variable by using pointer ). The result is implementation-defined if an attempt is made to change a const.

## 1) Pointer to variable.

---

C

```
int *ptr;
```

We can change the value of ptr and we can also change the value of object ptr pointing to. Pointer and value pointed by pointer both are stored in the read-write area. See the following code fragment.

---

C

```
#include <stdio.h>
int main(void)
{
    int i = 10;
    int j = 20;
    int *ptr = &i;
    /* pointer to integer */
}
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Couchbase's DBaaS is flexible, fast, familiar, free to try, and futureproof. Start a test drive today.

TRY NOW!

HIDE AD • AD VIA BUYSSELLADS

# Start Your Coding Journey Now!

[Login](#)[Register](#)

```
    return 0;  
}
```

## Output:

```
*ptr: 10  
*ptr: 20  
*ptr: 100
```

## 2) Pointer to constant.

Pointer to constant can be declared in following two ways.

---

### C

```
const int *ptr;
```

or

---

### C

```
int const *ptr;
```

We can change the pointer to point to any other integer variable, but cannot change the value of the object (entity) pointed using pointer ptr. The pointer is stored in the read-write area (stack in the present case). The object pointed may be in the read-only or read-write area. Let us see the following examples

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Couchbase's DBaaS is flexible, fast, familiar, free to try, and futureproof. Start a test drive today.

[TRY NOW!](#)

HIDE AD • AD VIA BUYSPELLADS

## Start Your Coding Journey Now!

[Login](#)[Register](#)

```
const int *ptr = &i;

printf("ptr: %d\n", *ptr);
/* error: object pointed cannot be modified
using the pointer ptr */
*ptr = 100;

ptr = &j;          /* valid */
printf("ptr: %d\n", *ptr);

return 0;
}
```

### Output:

error: assignment of read-only location ‘\*ptr’

Following is another example where variable i itself is constant.

## C

```
#include <stdio.h>

int main(void)
{
    /* i is stored in read only area*/
    int const i = 10;
    int j = 20;

    /* pointer to integer constant. Here i
    is of type "const int", and &i is of
    type "const int *". And p is of type
    "const int", types are matching no issue */
    int const *ptr = &i;
}
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Couchbase's DBaaS is flexible, fast, familiar, free to try, and futureproof. Start a test drive today.

[TRY NOW!](#)

HIDE AD • AD VIA BUYSSELLADS

## Start Your Coding Journey Now!

[Login](#)[Register](#)

```
ptr = &j;
printf("ptr: %d\n", *ptr);

return 0;
}
```

### Output:

error: assignment of read-only location ‘\*ptr’

Down qualification is not allowed in C++ and may cause warnings in C. Following is another example with down qualification.

## C

```
#include <stdio.h>

int main(void)
{
    int i = 10;
    int const j = 20;

    /* ptr is pointing an integer object */
    int *ptr = &i;

    printf("*ptr: %d\n", *ptr);

    /* The below assignment is invalid in C++, results in error
       In C, the compiler *may* throw a warning, but casting is
       implicitly allowed */
    ptr = &j;

    /* In C++, it is called 'down qualification'. The type of expression
       &i is "const int *" and the type of ptr is "int *" The
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Couchbase's DBaaS is flexible, fast, familiar, free to try, and futureproof. Start a test drive today.

[TRY NOW!](#)

HIDE AD • AD VIA BUYSSELLADS

# Start Your Coding Journey Now!

[Login](#)[Register](#)

```
printf("*ptr: %d\n", *ptr);

return 0;
}

// Reference:
// http://www.dansaks.com/articles/1999-02%20const%20T%20vs%20T%20const.pd
// More interesting stuff on C/C++ @ http://www.dansaks.com/articles.shtml
```

### 3) Constant pointer to variable.

#### C

```
int *const ptr;
```

Above declaration is a constant pointer to an integer variable, means we can change the value of object pointed by pointer, but cannot change the pointer to point another variable.

#### C

```
#include <stdio.h>

int main(void)
{
    . . . . .
}
```



[Data Structures](#) [Algorithms](#) [Interview Preparation](#) [Topic-wise Practice](#) [C++](#) [Java](#) [Python](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Couchbase's DBaaS is flexible, fast, familiar, free to try, and futureproof. Start a test drive today.

[TRY NOW!](#)

[HIDE AD](#) • [AD VIA BUYSSELLADS](#)

# Start Your Coding Journey Now!

[Login](#)[Register](#)

Output:

```
error: assignment of read-only variable 'ptr'
```

## 4) constant pointer to constant

### C

```
const int *const ptr;
```

Above declaration is a constant pointer to a constant variable which means we cannot change value pointed by the pointer as well as we cannot point the pointer to other variables. Let us see with an example.

### C

```
#include <stdio.h>

int main(void)
{
    int i = 10;
    int j = 20;
    /* constant pointer to constant integer */
    const int *const ptr = &i;

    printf("ptr: %d\n", *ptr);

    ptr = &j;      /* error */
    *ptr = 100;    /* error */

    return 0;
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Couchbase's DBaaS is flexible, fast, familiar, free to try, and futureproof. Start a test drive today.

[TRY NOW!](#)

HIDE AD • AD VIA BUYSSELLADS

# Start Your Coding Journey Now!

[Login](#)
[Register](#)

## error: assignment of read-only location 'ptr'

Summary :

Type	Declaration	pointer value change ( *ptr = 100 )	pointing value change ( ptr = &a)
1) Pointer to Variable	int * ptr	yes	yes
2) Pointer to Constant	<ul style="list-style-type: none"> <li>const int * ptr</li> <li>int const * ptr</li> </ul>	no	yes
3) Constant Pointer to Variable	int * const ptr	yes	no
4) Constant Pointer to Constant	const int * const ptr	no	no

This article is compiled by "**Narendra Kangralkar**". Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

If (Coding)

```
{
```

C foundation course = true;

Wait no more!



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Couchbase's DBaaS is flexible, fast, familiar, free to try, and futureproof. Start a test drive today.

[TRY NOW!](#)

HIDE AD • AD VIA BUYSSELLADS

# Start Your Coding Journey Now!

[Login](#)[Register](#)[Like](#) 1 / 4[Previous](#)[Next](#)

## RECOMMENDED ARTICLES

Page : 1 2

- 01** **Difference between const char \*p, char \* const p and const char \* const p**  
24, Jun 17
- 02** **Difference between const int\*, const int \* const, and int const \***  
19, Dec 18
- 03** **How does Volatile qualifier of C works in Computing System**  
28, Jul 20
- 04** **Understanding "volatile" qualifier in C | Set 1 (Introduction)**  
04, Dec 16
- 05** **Understanding "volatile" qualifier in C | Set 2 (Examples)**  
29, Nov 11
- 06** **Function overloading and const keyword**  
02, Oct 12
- 07** **Difference between #define and const in C?**  
29, Jul 14
- 08** **C++ | const keyword | Question 1**  
27, Aug 13

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Couchbase's DBaaS is flexible, fast, familiar, free to try, and futureproof. Start a test drive today.

[TRY NOW!](#)[HIDE AD](#) • [AD VIA BUYSSELLADS](#)



## Start Your Coding Journey Now!

[Login](#)[Register](#)[Easy](#)[Normal](#)[Medium](#)[Hard](#)[Expert](#)

Improved By : [NiranjanBSubramanian](#), [akshaynale115](#), [avinashsoni](#)

Article Tags : [C-Storage Classes and Type Qualifiers](#), [pointer](#), [C Language](#)

[Improve Article](#)[Report Issue](#)

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

[Load Comments](#)

5th Floor, A-118,  
Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

### Company

[About Us](#)[Careers](#)[In Media](#)

### Learn

[Algorithms](#)[Data Structures](#)[SDE Cheat Sheet](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge



Couchbase's DBaaS is flexible, fast, familiar, free to try, and futureproof. Start a test drive today.

[TRY NOW!](#)

HIDE AD • AD VIA BUYSSELLADS

# Start Your Coding Journey Now!

Login

Register

- Work & Career
- Business
- Finance
- Lifestyle

- CPP
- Golang
- C#
- SQL

## Web Development


- Web Tutorials
- Django Tutorial
- HTML
- CSS
- JavaScript
- Bootstrap

## Contribute

- Write an Article
- Improve an Article
- Pick Topics to Write
- Write Interview Experience
- Internships
- Video Internship

@geeksforgeeks , Some rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge

 **Couchbase**

Couchbase's DBaaS is flexible, fast, familiar, free to try, and futureproof. Start a test drive today.

TRY NOW!

HIDE AD • AD VIA BUYSSELLADS