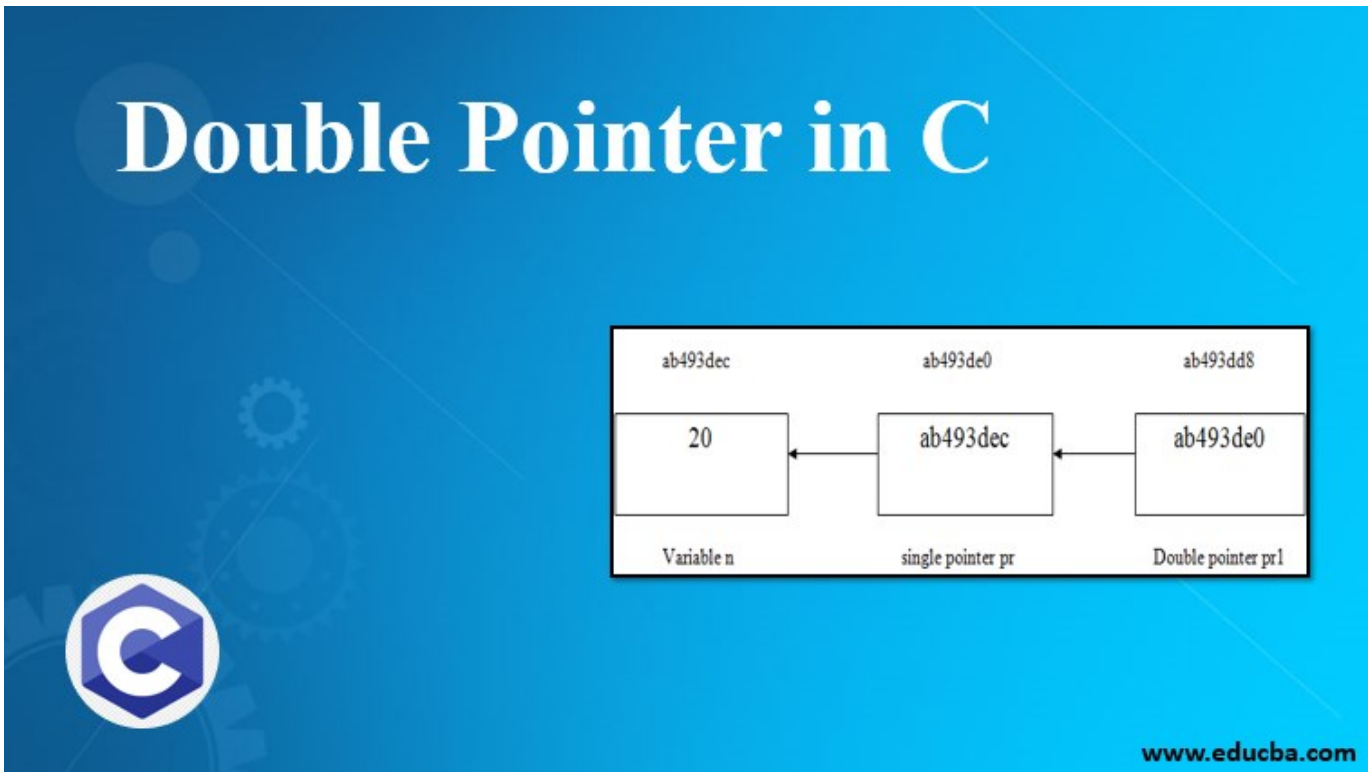
EDUCBA
(https://www.educba.com/software-development/)

← (https://www.educba.com/function-pointer-in-c/)

→ (https://www.educba.com/void-pointer-in-c/)




# Introduction to Double Pointer in C

In the C programming language, we have seen what pointers are and what they are use
C, a pointer means pointing directly to another variable. In general, Pointers are the variables that store the address of another variable. Whereas pointer to pointer which means a pointer

(https://www.educba.com/software-development/)

# How does Double Pointer work in C?

In this article, we will see how to declare double-pointer with syntax and example, and also we will see how to use them in the C programming language. So let us start from the syntax.

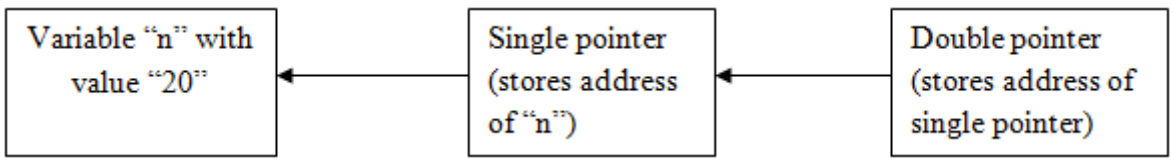


**Syntax:**

```
int **pointer_var;
```

In the above syntax, we can see the variable pointer_var is prefixed with two stars (**), also known as the indirection operator (*) for declaring the double-pointer.

So in general if the pointer is pointing to or referring to an object in memory then double-pointer is a pointer that would be pointing to or referring to another point where it is pointing to an object in memory. Let us see how this exactly works by below example and pictorial form:




# Examples to Implement Double Pointer in C

Below are the examples are mentioned:

## Example #1

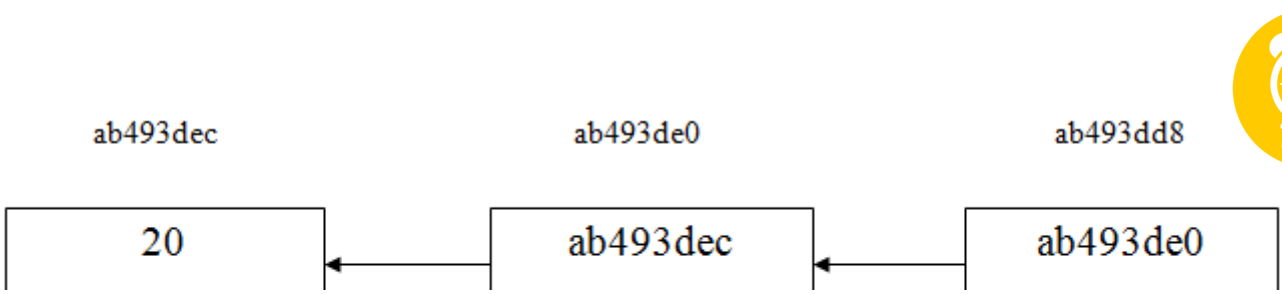
(https://www.educba.com/software-development/)

```
int n = 20;
int *pr;
int **pr1;
printf("\nThe address of the variable n is: %x\n", &n);
pr = &n;
printf("\nThe address of variable n stored in single pointer is: %x\n",pr);
pr1 = &pr;
printf("\nThe address of pointer pr stored in double pointer is: %x\n",pr1);
printf("\nThe address of double pointer pr1 is: %x\n", &pr1);
printf("\nThe value stored at pointer pr: %d\n",*pr);
printf("\nThe value stored at another pointer pr1: %d\n",**pr1);
}
```

**Output:**

```
The address of the variable n is: ab493dec
The address of variable n stored in single pointer is: ab493dec
The address of pointer pr stored in double pointer is: ab493de0
The address of double pointer pr1 is: ab493dd8
The value stored at pointer pr: 20
The value stored at another pointer pr1: 20
```

**Explanation:** In the above code, we have declared a variable “n” and initialized it to valu now we have declared a single pointer “*pr” and double pointer “**pr1” where the address of variable n will be stored in pointer”*pr” and the address of this single pointer “*pr” is stored in



(https://www.educba.com/software-development/)

In this article, let us see why and where double pointers can be used. There are several uses of a pointer to pointer where it is the address of a data. This can be explained by writing this code.

**Code:**

```
int n = 50;
int* p1 =&n;
int** p2= &p1;
```

If we see the above code if “n” is at the address 100 and pointer “p1” is pointing or assigned to the address of n (100) and p1 also has address 200 and pointer “p2” is now assigned to the address of p1 (200). Another use of a double pointer is when we want to allocate space in the matrix. This can be explained in the below code.

**Code:**

```
#include <stdlib.h>
int main(){
int **matrix;
int row=5,col=5;
int i;
matrix = (int**)malloc(row*sizeof(int*));


for (i=0;i<row;i++)
matrix[i]= (int*)malloc(col*sizeof(int));
}
```


(https://www.educba.com/software-development/)

the address of the "matrix" pointer is an integer. Therefore, in the same way, a pointer to an integer pointer can have the starting address in the array of an integer as that is also an integer.

## Example #3

Double pointers can also be used when we want to alter or change the value of the pointer. In general double pointers are used if we want to store or reserve the memory allocation or assignment even outside of a function call we can do it using double pointer by just passing these functions with ** arg. As we did it in the previous code. Let us consider an example where we want to change or update a character from a function.

**Code:**

```
void func(char ch)
{
ch = 'B';
}
int main()
{
char ptr;
ptr = 'A';
printf("%c", ptr);



func(ptr);
printf("%c\n", ptr);
}
```

(https://www.educba.com/software-development/)

```
#include<stdio.h>
void func( char *p)
{
*p = 'Y';
}
int main()
{
char *p;
p = (char *)malloc(sizeof(char) * 1);
*p = 'X';
printf("%c\n", *p);
func(p);
printf("%c\n", *p);
}
```

**Output:**

```
X
Y
```

**Explanation:** So in the above code, it will allow you to update the character at the pointer “p” with value “X” to the value “Y”.


## Conclusion

In this article, we can conclude that pointers are also variables that can store values. But a pointer usually stores the value as the address of another variable. So commonly we can define

(https://www.educba.com/software-development/)

## Recommended Article

This is a guide to Double Pointer in C. Here we discuss how Double Pointer works in C and examples for better understanding. You can also go through our other related articles to learn more –

1. Pointers in C (https://www.educba.com/pointers-in-c/)
2. Nested if Statement in C (https://www.educba.com/nested-if-statement-in-c/)
3. C Literals (https://www.educba.com/c-literals/)
4. Void Pointer in C (https://www.educba.com/void-pointer-in-c/)

(https://www.educba.com/software-development/)

## About Us

Blog (https://www.educba.com/blog/?source=footer)

Who is EDUCBA? (https://www.educba.com/about-us/?source=footer)

Sign Up (https://www.educba.com/software-development/signup/?source=footer)

Corporate Training (https://www.educba.com/corporate/?source=footer)

Certificate from Top Institutions (https://www.educba.com/educbalive/?source=footer)

Contact Us (https://www.educba.com/contact-us/?source=footer)

Verifiable Certificate (https://www.educba.com/software-development/verifiable-certificate/?source=footer)

Reviews (https://www.educba.com/software-development/reviews/?source=footer)

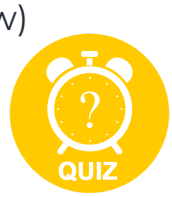
Terms and Conditions (https://www.educba.com/terms-and-conditions/?source=footer)

Privacy Policy (https://www.educba.com/privacy-policy/?source=footer)

## Apps

iPhone & iPad (https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8)

Android (https://play.google.com/store/apps/details?id=com.educba.www)




## Resources

Free Courses (https://www.educba.com/software-development/free-courses/?source=footer)

(https://www.educba.com/software-development/)

## Certification Courses

All Courses (https://www.educba.com/software-development/courses/?source=footer)

Software Development Course - All in One Bundle (https://www.educba.com/software-development/courses/software-development-course/?source=footer)

Become a Python Developer (https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Java Course (https://www.educba.com/software-development/courses/java-course/?source=footer)

Become a Selenium Automation Tester (https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer)

Become an IoT Developer (https://www.educba.com/software-development/courses/iot-course/?source=footer)

ASP.NET Course (https://www.educba.com/software-development/courses/asp-net-course/?source=footer)

VB.NET Course (https://www.educba.com/software-development/courses/vb-net-course/?source=footer)

PHP Course (https://www.educba.com/software-development/courses/php-course/?source=footer)