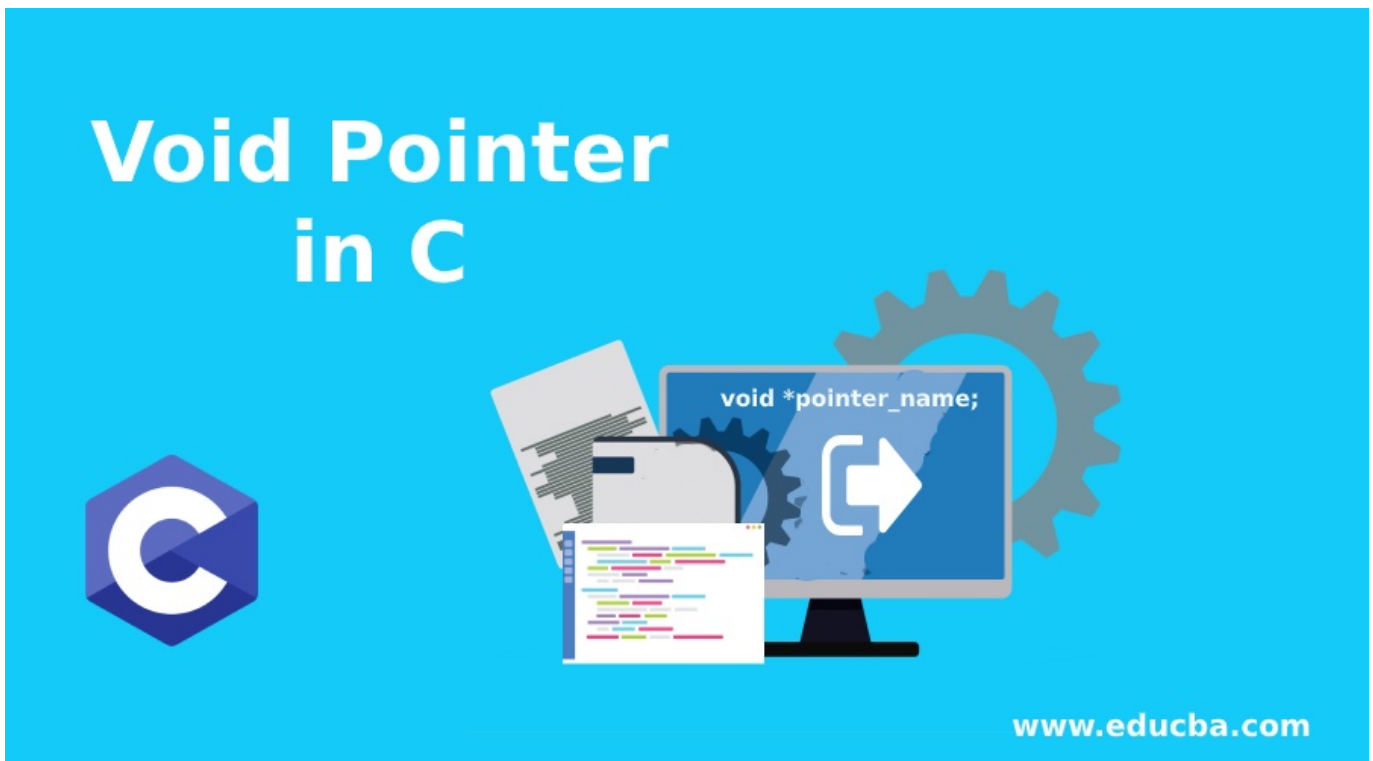




[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

← (<https://www.educba.com/double-pointer-in-c/>)

→ (<https://www.educba.com/const-pointer-in-c/>)



## Definition of C Void Pointer

---

A void pointer in C is a pointer that does not have any associated data type. A void pointer clearly indicates that it is empty and can only be capable of holding the addresses of any type. Further, these void pointers with addresses can be typecast into any other type easily. Memory





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

### **Start Your Free Software Development Course**

Web development, programming languages, Software testing & others

```
void *pointer_name;
```

The syntax flow follows in a way that keyword void is the type of pointer followed by the name of the pointer which is being pointed and allocated as an address allocation. The Pointer declaration is performed with the pointer name and the pointer type supporting any data type. Representation of pointer in terms of C is the same as the pointer of character type.

#### **Example:**

```
void *ptr
```

This example shows that the pointer is expecting a void type of pointer and then it is being pointed by the pointer whose name is given as ptr inclusive of '\*' symbol which denotes that a pointer is being declared and will be used in mere future for dereferencing purpose.

## **How does Void Pointer Work in C?**

The pointer concept in C is very useful as it helps in memory allocation and address management. It helps in implementing two types of pointers namely void pointers and generic pointers. Therefore, it is sometimes called a general-purpose pointer. Referencing and





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

mentioned earlier referencing and dereferencing are some of the methods associated with pointer concept which will be used. Dereferencing comes into picture whenever it is a need to access the stored value in the pointer variable. Also, there is a type of casting value which is used for dereferencing because none of the pointer value is associated with the data types. The compiler also cannot find the type of variable which is pointed by any type of void pointer. One point to keep in mind is void pointer will not support any kind of arithmetic operation. It makes use of indirection operator '\*' to serve the entire purpose. But to serve this problem there is a need to typecast the pointer variable as well for dereferencing. The usage of typecasting is needed because there is no presence of datatype associated at the time of declaration of the pointer. In short, the compiler doesn't have any appropriate source to get an idea of the type of data type declared. So, it performs the typecasting and meets the requirement to give an intimation of the type of data type used by the void pointer at the time of declaration.

Size of the void pointer is the next point of focus as a void pointer in C functions almost the same as character pointer in C which means a representation of Character type of pointer will be the same as a void pointer in C. Also, the size will vary according to the platform being used by the pointer. Memory allocation also works in some format which means void pointer has the beauty of providing an enhancement feature of memory management with `calloc ()` and `malloc ()` functions which ultimately returns the void as return type. Therefore, these functions can be used to allocate the memory of any data type.

🔗 Popular Course in this category





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

[View Course](https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1)

(<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1>)

#### Related Courses

C++ Training (4 Courses, 5 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1>)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)

The most important theme line of all the added advantage which pointer has is that it has the power of reusability for void pointers. It can store any type of object and can retrieve any type of object from the defined object using the indirection operator and operator with proper typecasting. Dereferencing operator as part of the pointer can be used for easy access or manipulation of the stored data in the memory location for the pointer pointing the data type and it will have a direct impact on the value of the data type.

## Examples of Void Pointer in C

Following are the examples as given below:

### Example #1

This program illustrated the void Pointer in C as it is not associated with any data type at the time of declaration as shown in the given output.

Code:

```
int r = 11;
```





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

Output:

```
main.c:4:1: warning: data definition has no type or storage class
p = &m;
^
main.c:4:1: warning: type defaults to 'int' in declaration of 'p' [-Wimplicit-int]
main.c:4:1: error: conflicting types for 'p'
main.c:3:7: note: previous definition of 'p' was here
void *p = &r;
^
main.c:4:5: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
p = &m;
^
main.c:4:5: error: initializer element is not computable at load time
```

## Example #2

This program is used to illustrate the dereferencing of the void pointer of C where the input is given to variable with the inception operator which is shown with the following example.

Code:

```
#include<stdio.h>

int main()
{
    int o = 18;
    void *ptr = &o;

    printf("%d", *(int *)ptr);
    return 0;
}
```





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

## Example #3

This program illustrates the void pointer in C for representing the size of integer value define with the arithmetic operator which means it manages to support the arithmetic operator in association with a void pointer. Although it needs to follow some of the standards of the GNU then only the compilation allowed will provide the necessary output as shown otherwise it will give a compilation error.

### Code:

```
#include<stdio.h>
int main()
{
    int q[3] = {5,9,7};
    void *ptr = &q;
    ptr = ptr + sizeof(int);
    printf("%d", *(int *)ptr);
    return 0;
}
```

### Output:

9

## Example #4

This program is used to illustrate the basic concept of dereferencing with the void pointer in C and its output is as follows.





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

```
void main(),
{
    int r=17;
    float q=13.8;
    char i='c';
    void *p;
    p=&r;
    printf("%d",*((int*)p));
    p=&q;
    printf("n%f",*((float*)p));
    p=&i;
    printf("n%c",*((char*)p));
}
```

Output:

```
17n13.800000nc
```

## Conclusion

void pointer in C is used to mitigate the problem of pointers pointing to each other with a different set of values and data types. There is also a reduction in explicit typecasting. Also, it supports the generic pointer type which makes it as a generic-purpose compiler.

## Recommended Articles



This is a guide to Void Pointer in C. Here we also discuss the definition and how does void pointer work in c? along with different examples and code implementation. You may also have a



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

3. [Function Pointer in C \(https://www.educba.com/function-pointer-in-c/\)](https://www.educba.com/function-pointer-in-c/)
4. [Null pointer in C \(https://www.educba.com/null-pointer-in-c/\)](https://www.educba.com/null-pointer-in-c/)

## C PROGRAMMING TRAINING (3 COURSES, 5 PROJECT)

- ☒ 3 Online Courses
- ☒ 5 Hands-on Projects
- ☒ 34+ Hours
- ☒ Verifiable Certificate of Completion
- ☒ Lifetime Access

**Learn More**

<https://www.educba.com/software-development/courses/c-programming-course/?btnz=educba-inline-banner3>

---

**About Us**

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







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

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>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)

## Certification Courses





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

[development/courses/python-certification-course/?source=footer\)](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>)

© 2020 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

