









Winter Tickets Sale!

C++ Data Types C++ Input/Output C++ Arrays C++ Pointers C++ OOPs C++ STL C++ Interview Questions C++ Programs

160 Days of DSA

Share Your Experiences

reinterpret_cast in C++ | Type Casting operators

const_cast in C++ | Type Casting operators

Casting Operators in C++

C/C++ Ternary Operator - Some Interesting Observations

Types of Operator Overloading in C++

Default Assignment Operator and References in C++

reinterpret_cast in C++ | Type **Casting operators**

Last Updated: 30 May, 2022









reinterpret_cast is a type of casting operator used in C++.

- It is used to convert a pointer of some data type into a pointer of another data type, even if the data types before and after conversion are different.
- It does not check if the pointer type and data pointed by the pointer is same or not.

Syntax:

```
data type *var name =
       reinterpret cast <data type *>
(pointer variable);
```

Function Overloading and Return Type in

Return Type

• It doesn't have any return type. It simply converts the pointer type.

Parameters

• It takes only one parameter i.e., the source pointer variable (p in above example).

CPP

```
// CPP program to demonstrate working of
// reinterpret_cast
#include <iostream>
using namespace std;

int main()
{
    int* p = new int(65);
    char* ch = reinterpret_cast<char*>(p);
    cout << *p << endl;
    cout << p << endl;
    cout << p << endl;
    cout << p << endl;
    cout << ch << endl;</pre>
```

```
return 0;
}
```

Output:

```
65
A
0x1609c20
A
```

Purpose for using reinterpret_cast

- 1. reinterpret_cast is a very special and dangerous type of casting operator. And is suggested to use it using proper data type i.e., (pointer data type should be same as original data type).
- 2. It can typecast any pointer to any other data type.
- 3. It is used when we want to work with bits.
- 4. If we use this type of cast then it becomes a nonportable product. So, it is suggested not to use this concept unless required.
- 5. It is only used to typecast any pointer to its original type.

 Skip to content

6. Boolean value will be converted into integer value i.e., 0 for false and 1 for true.

CPP

```
\bigcirc // CPP code to illustrate using structure
    #include <bits/stdc++.h>
    using namespace std;
    // creating structure mystruct
    struct mystruct {
        int x;
        int y;
        char c;
        bool b;
    };
    int main()
        mystruct s;
        // Assigning values
        s.x = 5;
        s.y = 10;
        s.c = 'a';
        s.b = true;
        // data type must be same during casting
        // as that of original
```

```
// converting the pointer of 's' to,
// pointer of int type in 'p'.
int* p = reinterpret cast<int*>(&s);
cout << sizeof(s) << endl;</pre>
// printing the value currently pointed by *r
cout << *p << endl;</pre>
// incrementing the pointer by 1
p++;
// printing the next integer value
cout << *p << endl;</pre>
p++;
// we are casting back char * pointed
// by p using char *ch.
char* ch = reinterpret cast<char*>(p);
// printing the character value
// pointed by (*ch)
cout << *ch << endl;</pre>
ch++;
/* since, (*ch) now points to boolean value,
so it is required to access the value using
same type conversion.so, we have used
data type of *n to be bool. */
bool* n = reinterpret_cast<bool*>(ch);
cout << *n << endl;</pre>
                     Skip to content
```

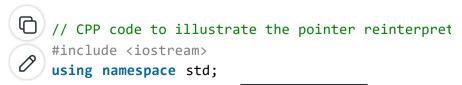
```
// we can also use this line of code to
// print the value pointed by (*ch).
cout << *(reinterpret_cast<bool*>(ch));
return 0;
}
```

Output:

```
12
5
10
a
1
```

Program 2

CPP



```
class A {
    public:
        void fun a()
-;0;-
            cout << " In class A\n";</pre>
    };
    class B {
    public:
        void fun_b()
            cout << " In class B\n";</pre>
    };
    int main()
        // creating object of class B
        B* x = new B();
        // converting the pointer to object
        // referenced of class B to class A
        A* new_a = reinterpret_cast<A*>(x);
        // accessing the function of class A
        new_a->fun_a();
        return 0;
```

Output:

In class A

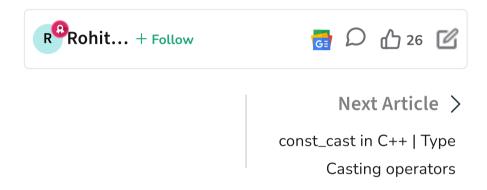
Related link:

https://www.geeksforgeeks.org/casting-operators-in-c-set-1-const_cast/
https://stackoverflow.com/questions/573294/when-to-use-reinterpret-cast" rel="noopener"
target="_blank
http://forums.codeguru.com/showthread.php?482227-reinterpret_cast-lt-gt-and-where-can-it-be-used
https://www.ibm.com/support/knowledgecenter/en/SS
LTBW_2.3.0/com.ibm.zos.v2r3.cbclx01/keyword_reint
erpret_cast.htm
https://stackoverflow.com/questions/573294/when-to-use-reinterpret-cast

Learn in a distraction-free environment with refined, high-quality content and 35+ expert-led tech courses to help you crack any interview. From programming languages and DSA to web development and data science, <u>GeeksforGeeks</u>

Premium has you covered!

Choose <u>GeeksforGeeks Premium</u> today and also get access to **Unlimited Article Summarization**, **100% Ad free environment**, **A.I. Bot support** in all coding problems, and much more. <u>Go Premium!</u>



Similar Reads

const_cast in C++ | Type Casting operators

C++ supports following 4 types of casting operators: 1. const_cast 2. static_cast 3. dynamic_cast 4....

(4 min read

Casting Operators in C++

Casting operators are used for type casting in C++.

They are used to convert one Skip to content other. C+...

(5 min read

unordered_set operators in C++ STL

Unordered_set provides two operators in C++ STL. These are: Syntax: 1. (unordered_set &lhs ==...

(5 min read

C++ Pointer Operators

Prerequisite: Pointers in C++ A pointer variable is a variable that stores the address of another variable or ...

(2 min read

C++ Arithmetic Operators

Arithmetic Operators in C++ are used to perform arithmetic or mathematical operations on the operand...

(3 min read

Input/Output Operators Overloading in C++

Operator Overloading is a part of Polymorphism, which enables the feature because of which we can directly...

(2 min read

C++ Increment and Decrement Operators

Prerequisite: Operators in C++ What is a C++ increment Operator? The C++ increment operator is a unary...

(4 min read

C++ Comparison Operators

Comparison operators are operators used for comparing two elements, these are mostly used with if-else...

(3 min read

C++ Relational Operators

In C++ programming language, we sometimes require to compare values and expressions. This comparison...

(4 min read

C++ Logical Operators

In C++ programming languages, logical operators are symbols that allow you to combine or modify conditio...

(4 min read

Unary Operators In C++

Unary operators in C++ are those operators that work on a single value (operand). They perform operations...

(8 min read

Assignment Operators In C++

In C++, the assignment operator forms the backbone of many algorithms and computational processes by...

(7 min read

What are the Operators that Can be and Cannot be...

There are various ways to overload Operators in C++ by implementing any of the following types of functions: ...

(4 min read

Increment (Decrement) operators require L-value...

What will be the output of the following program? #include<stdio.h> int main() { int i = 10; printf("%d", +...

(1) 1 min read

Order of operands for logical operators

The order of operands of logical operators &&, || are important in C/C++. In mathematics, logical AND, OR,...

(1 min read

Overloading stream insertion (<>) operators in C++

In C++, stream insertion operator "<<" is used for output and extraction operator ">>" is used for input. We mus...

(2 min read

How to sum two integers without using arithmetic...

Given two integers a and b, how can we evaluate the sum a + b without using operators such as +, -, ++, --,...

(4 min read

Conditionally assign a value without using...

Given 4 integers a, b, y, and x, where x can assume the values of either 0 or 1 only. The following question is...

(6 min read

Written version of Logical operators in C++

Can we use keywords in place of operators in C++? Yes, certainly, we can. The ANSI C++ Standard has propos...

(2 min read

Comparing String objects using Relational Operato...

If strings are compared using relational operators then, their characters are compared lexicographically...

(2 min read

Article Tags:

C++

cpp-advanced

cpp-pointer

Practice Tags:





Ocrporate & Communications Address:-A-143, 9th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305) | Registered Address:-K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305















Company	Explore	Languages	DSA	Data Science	Web
About Us	Job-A-Thon	Python	Data Structures	& ML	Technologies
About Us Legal Careers In Media Contact Us Advertise with us GFG Corporate Solution Placement Training Program	Job-A-Thon Hiring Challenge Hack-A-Thon GfG Weekly Contest Offline Classes (Delhi/NCR) DSA in JAVA/C++ Master System Design Master CP	Python Java C++ PHP GoLang SQL R Language Android Tutorial	Data Structures Algorithms DSA for Beginners Basic DSA Problems DSA Roadmap DSA Interview Questions Competitive Programming	& ML Data Science With Python Data Science For Beginner Machine Learning ML Maths Data Visualisation Pandas	Technologies HTML CSS JavaScript TypeScript ReactJS NextJS NodeJS Bootstrap Tailwind CSS
	GeeksforGeeks Videos			NumPy NLP	
	Geeks Community			Deep Learning	

Python					
Tutorial					
Python					
Programming					
Examples					
Django Tutorial					
Python Projects					
Python Tkinter					
Web Scraping					
OpenCV Tutorial					

Computer
Science
GATE CS Notes
Operating
Systems
Computer
Network
Database
Management
System

DevOps
Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps
Roadmap

Design
High Level
Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design

Bootcamp

System

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science

English Grammar

Commerce

Accountancy
Business Studies
Economics
Management
HR Management
Finance
Income Tax

Python Interview Software Interview

Question Engineering Questions

Digital Logic

Design

Engineering

Maths

Databases	Preparation	Competitive	More	Free Online	Write & Earn
SQL	Corner	Exams	Tutorials	Tools	Write an Article
MYSQL PostgreSQL PL/SQL MongoDB	Company-Wise Recruitment Process Resume Templates Aptitude Preparation Puzzles Company-Wise Preparation Companies	JEE Advanced UGC NET UPSC SSC CGL SBI PO SBI Clerk IBPS PO IBPS Clerk	Software Development Software Testing Product Management Project Management Linux Excel All Cheat Sheets Recent Articles	Typing Test Image Editor Code Formatters Code Converters Currency Converter Random Number Generator Random Password Generator	Improve an Article Pick Topics to Write Share your Experiences Internships
	Colleges				

DSA/Placements Development/Testflaghine

JavaScript Full DSA - Self Paced Learning/Data Course Course Science React JS Course DSA in Complete JavaScript - Self React Native Machine Paced Course Course Learning & Data DSA in Python -Django Web Science Program Self Paced Development - [LIVE] Course Complete

Programming Languages

C Programming
with Data
Structures
C++
Programming
Course

Clouds/Devops GATE

DevOps GATE CS & IT Test Engineering Series - 2025 **AWS Solutions GATE DA Test** Architect Series 2025 Certification GATE CS & IT Salesforce Course - 2025 Certified **GATE DA Course** Administrator 2025 Course

reinterpret_cast in C++ | Type Casting operators - GeeksforGeeks

reinterpret_cast in C++ Type Casting operators - GeekstorGeeks								
C Programming	Full Stack	Data Analytics	Java					
Course Online -	Development -	Training using	Programming					
Learn C with	[LIVE]	Excel, SQL,	Course					
Data Structures	JAVA Backend	Python &	Python Full					
Complete	Development -	PowerBI - [LIVE]	Course					
Interview	[LIVE]	Data Science						
Preparation	Complete	Training Program						
Master	Software Testing	- [LIVE]						
Competitive	Course [LIVE]	Mastering						
Programming	Android Mastery	Generative Al						
Core CS Subject	with Kotlin [LIVE]	and ChatGPT						
for Interview								
Preparation								
Mastering								
System Design:								
LLD to HLD								
Tech Interview								
101 - From DSA								
to System Design								
[LIVE]								
DSA to								
Development								
[HYBRID]								
Placement								
Preparation								
Crash Course								
[LIVE]								

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved