

# Toán tử trong C++

Một toán tử là một biểu tượng, mà nói cho compiler thực hiện các thao tác toán học và logic cụ thể. C++ cung cấp nhiều toán tử có sẵn, đó là:

- Toán tử số học
- Toán tử quan hệ
- Toán tử logic
- Toán tử so sánh bit
- Toán tử gán
- Toán tử hỗn hợp

## Toán tử số học trong C++

Bảng dưới liệt kê các toán tử số học được hỗ trợ bởi ngôn ngữ C++:

Giả sử biến A giữ giá trị 10, biến B giữ 20 thì:

### Ví dụ toán tử số học

Toán tử	Miêu tả	Ví dụ
+	Cộng hai toán hạng	$A + B$ kết quả là 30
-	Trừ toán hạng thứ hai từ toán hạng đầu	$A - B$ kết quả là -10
*	Nhân hai toán hạng	$A * B$ kết quả là 200

/	Phép chia	B / A kết quả là 2
%	Phép lấy số dư	B % A kết quả là 0
++	<b>Toán tử tăng (++)</b> , tăng giá trị toán hạng thêm một đơn vị	A++ kết quả là 11
--	<b>Toán tử giảm (--)</b> , giảm giá trị toán hạng đi một đơn vị	A-- kết quả là 9

## Toán tử quan hệ trong C++

Bảng dưới đây liệt kê các toán tử quan hệ được hỗ trợ bởi ngôn ngữ C++:

Giả sử biến A giữ giá trị 10, biến B giữ 20 thì:

### Ví dụ toán tử quan hệ

Toán tử	Miêu tả	Ví dụ
==	Kiểm tra nếu 2 toán hạng bằng nhau hay không. Nếu bằng thì điều kiện là true.	(A == B) là không đúng
!=	Kiểm tra 2 toán hạng có giá trị khác nhau hay không. Nếu không bằng thì điều kiện là true.	(A != B) là true

>	Kiểm tra nếu toán hạng bên trái có giá trị lớn hơn toán hạng bên phải hay không. Nếu lớn hơn thì điều kiện là true.	$(A > B)$ là không đúng
<	Kiểm tra nếu toán hạng bên trái nhỏ hơn toán hạng bên phải hay không. Nếu nhỏ hơn thì là true.	$(A < B)$ là true
>=	Kiểm tra nếu toán hạng bên trái có giá trị lớn hơn hoặc bằng giá trị của toán hạng bên phải hay không. Nếu đúng là true.	$(A \geq B)$ là không đúng
<=	Kiểm tra nếu toán hạng bên trái có giá trị nhỏ hơn hoặc bằng toán hạng bên phải hay không. Nếu đúng là true.	$(A \leq B)$ là true

## Toán tử logic trong C++

Bảng dưới đây chỉ rõ tất cả các toán tử logic được hỗ trợ bởi ngôn ngữ C.

Giả sử biến A có giá trị 1 và biến B có giá trị 0:

### Ví dụ toán tử logic

Toán tử	Miêu tả	Ví dụ

&&	Được gọi là toán tử logic AND (và). Nếu cả hai toán tử đều có giá trị khác 0 thì điều kiện trở lên true.	(A && B) là false.
	Được gọi là toán tử logic OR (hoặc). Nếu một trong hai toán tử khác 0, thì điều kiện là true.	(A    B) là true.
!	Được gọi là toán tử NOT (phủ định). Sử dụng để đảo ngược lại trạng thái logic của toán hạng đó. Nếu điều kiện toán hạng là true thì phủ định nó sẽ là false.	!(A && B) là true.

## Toán tử so sánh bit trong C++

Toán tử so sánh bit làm việc trên đơn vị bit, tính toán biểu thức so sánh từng bit. Bảng dưới đây về &, |, và ^ như sau:

p	q	p & q	p   q	p ^ q
0	0	0	0	0
0	1	0	1	1
1	1	1	1	0
1	0	0	1	1

Giả sử nếu A = 60; và B = 13; thì bây giờ trong định dạng nhị phân chúng sẽ là như sau:

A = 0011 1100

B = 0000 1101

-----

A&B = 0000 1100

A|B = 0011 1101

A^B = 0011 0001

~A = 1100 0011

Các toán tử so sánh bit được hỗ trợ bởi ngôn ngữ C++ được liệt kê trong bảng dưới đây. Giả sử ta có biến A có giá trị 60 và biến B có giá trị 13, ta có:

### Ví dụ

Toán tử	Miêu tả	Ví dụ
&	Toán tử AND (và) nhị phân sao chép một bit tới kết quả nếu nó tồn tại trong cả hai toán hạng.	(A & B) sẽ cho kết quả là 12, tức là 0000 1100
	Toán tử OR (hoặc) nhị phân sao chép một bit tới kết quả nếu nó tồn tại trong một hoặc hai toán hạng.	(A   B) sẽ cho kết quả là 61, tức là 0011 1101

$\wedge$	Toán tử XOR nhị phân sao chép bit mà nó chỉ tồn tại trong một toán hạng mà không phải cả hai.	$(A \wedge B)$ sẽ cho kết quả là 49, tức là 0011 0001
$\sim$	Toán tử đảo bit (đảo bit 1 thành bit 0 và ngược lại).	$(\sim A)$ sẽ cho kết quả là -61, tức là 1100 0011.
$\ll$	Toán tử dịch trái. Giá trị toán hạng trái được dịch chuyển sang trái bởi số các bit được xác định bởi toán hạng bên phải.	$A \ll 2$ sẽ cho kết quả 240, tức là 1111 0000 (dịch sang trái hai bit)
$\gg$	Toán tử dịch phải. Giá trị toán hạng trái được dịch chuyển sang phải bởi số các bit được xác định bởi toán hạng bên phải.	$A \gg 2$ sẽ cho kết quả là 15, tức là 0000 1111 (dịch sang phải hai bit)

## Toán tử gán trong C++

Dưới đây là những toán tử gán được hỗ trợ bởi ngôn ngữ C++:

### Ví dụ

Toán tử	Miêu tả	Ví dụ
=	Toán tử gán đơn giản. Gán giá trị toán hạng bên phải cho toán hạng trái.	$C = A + B$ sẽ gán giá trị của $A + B$ vào trong C

+=	Thêm giá trị toán hạng phải tới toán hạng trái và gán giá trị đó cho toán hạng trái.	$C += A$ tương đương với $C = C + A$
-=	Trừ đi giá trị toán hạng phải từ toán hạng trái và gán giá trị này cho toán hạng trái.	$C -= A$ tương đương với $C = C - A$
*=	Nhân giá trị toán hạng phải với toán hạng trái và gán giá trị này cho toán hạng trái.	$C *= A$ tương đương với $C = C * A$
/=	Chia toán hạng trái cho toán hạng phải và gán giá trị này cho toán hạng trái.	$C /= A$ tương đương với $C = C / A$
%=	Lấy phần dư của phép chia toán hạng trái cho toán hạng phải và gán cho toán hạng trái.	$C \% = A$ tương đương với $C = C \% A$
<<=	Dịch trái toán hạng trái sang số vị trí là giá trị toán hạng phải.	$C <<= 2$ tương đương với $C = C << 2$
>>=	Dịch phải toán hạng trái sang số vị trí là giá trị toán hạng phải.	$C >>= 2$ tương đương với $C = C >> 2$
&=	Phép AND bit	$C \&= 2$ tương đương với $C = C \& 2$
^=	Phép OR loại trừ bit	$C \wedge= 2$ tương đương với $C = C \wedge 2$
=	Phép OR bit.	$C  = 2$ tương đương với $C = C   2$

## Các toán tử hỗn hợp trong C++

Dưới đây là một số toán tử hỗn hợp quan trọng được hỗ trợ bởi ngôn ngữ C++.

Toán tử	Miêu tả
sizeof	<b>Toán tử sizeof trong C++</b> trả về kích cỡ của một biến. Ví dụ: sizeof(a), với a là integer, sẽ trả về 4
Điều kiện ? X : Y	<b>Toán tử điều kiện trong C++</b> . Nếu Condition là true ? thì nó trả về giá trị X : nếu không thì trả về Y
,	<b>Toán tử Comma trong C++</b> làm cho một dãy hoạt động được thực hiện. Giá trị của toàn biểu thức comma là giá trị của biểu thức cuối cùng trong danh sách được phân biệt bởi dấu phẩy
. (dot) và -> (arrow)	<b>Toán tử thành viên trong C++</b> được sử dụng để tham chiếu các phần tử đơn của các lớp, các cấu trúc, và union
Cast	<b>Toán tử ép kiểu (Casting) trong C++</b> biến đổi một kiểu dữ liệu thành kiểu khác. Ví dụ: int(2.2000) sẽ trả về 2
&	<b>Toán tử con trỏ &amp; trong C++</b> trả về địa chỉ của một biến. Ví dụ: &a; sẽ trả về địa chỉ thực sự của biến này
*	<b>Toán tử con trỏ * trong C++</b> là trỏ tới một biến. Ví dụ: *var sẽ trỏ tới một biến var



## Thứ tự ưu tiên toán tử trong C++

Thứ tự ưu tiên toán tử trong C++ xác định cách biểu thức được tính toán. Ví dụ, toán tử nhân có quyền ưu tiên hơn toán tử cộng, và nó được thực hiện trước.

Ví dụ,  $x = 7 + 3 * 2$ ; ở đây,  $x$  được gán giá trị 13, chứ không phải 20 bởi vì toán tử  $*$  có quyền ưu tiên cao hơn toán tử  $+$ , vì thế đầu tiên nó thực hiện phép nhân  $3 * 2$  và sau đó thêm với 7.

Bảng dưới đây liệt kê thứ tự ưu tiên của các toán tử. Các toán tử với quyền ưu tiên cao nhất xuất hiện trên cùng của bảng, và các toán tử có quyền ưu tiên thấp nhất thì ở bên dưới cùng của bảng. Trong một biểu thức, các toán tử có quyền ưu tiên cao nhất được tính toán đầu tiên.

### Ví dụ

Loại	Toán tử	Thứ tự ưu tiên
Postfix	() [] -> . ++ --	Trái sang phải
Unary	+ - ! ~ ++ -- (type)* & sizeof	Phải sang trái
Tính nhân	* / %	Trái sang phải
Tính cộng	+ -	Trái sang phải
Dịch chuyển	<< >>	Trái sang phải

Quan hệ	< <= > >=	Trái sang phải
Cân bằng	== !=	Trái sang phải
Phép AND bit	&	Trái sang phải
Phép XOR bit	^	Trái sang phải
Phép OR bit		Trái sang phải
Phép AND logic	&&	Trái sang phải
Phép OR logic		Trái sang phải
Điều kiện	?:	Phải sang trái
Gán	= += -= *= /= %= >>= <<= &= ^=  =	Phải sang trái
Dấu phẩy	,	Trái sang phải