Basic Operations in C++

Các phép toán cơ bản trong C++ đề cập đến các phép toán số học, quan hệ và logic cơ bản có thể được thực hiện bằng ngôn ngữ lập trình C++, vốn rất cần thiết cho bất kỳ loại chương trình hoặc phép tính nào trong thực tế. Dưới đây là tóm tắt về các phép toán cơ bản trong C++.

***Arithmetic Operations (các phép tính toán học)***

Các phép toán này được sử dụng để thực hiện các phép tính trong C++ và bao gồm các phép tính sau:

Phép cộng (+): Cộng hai số.

int a = 5; int b = 6; int sum = a + b; // tổng bằng 11

Phép trừ (-): Trừ một số với số kia.

int a = 10; int b = 6; int diff = a - b; // diff bằng 4

Phép nhân (\*): Nhân hai số.

int a = 3; int b = 4; int product = a \* b; // tích bằng 12

Phép chia (/): Chia một số cho một số khác, cho kết quả là thương.

int a = 12; int b = 4; int quotient = a / b; // thương bằng 3

Môđun (%): Chia một số cho một số khác, cho kết quả là số dư.

int a = 15; int b = 4; int remainder = a % b; // số dư là 3

***Relational Operators (Toán tử quan hệ)***

Các phép toán này so sánh hai giá trị và trả về một giá trị boolean (true/false) tùy thuộc vào phép so sánh. Các phép toán quan hệ là:

Equal to (==): Trả về true nếu cả hai toán hạng bằng nhau.

5 == 5 // true 3 == 4 // false

Not equal to (!=): Trả về true nếu các toán hạng không bằng nhau.

5 != 2 // true 1 != 1 // false

Greater than (>): Trả về true nếu toán hạng thứ nhất lớn hơn toán hạng thứ hai.

5 > 3 // true 2 > 3 // false

Less than (<): Trả về true nếu toán hạng thứ nhất nhỏ hơn toán hạng thứ hai.

3 < 5 // true 6 < 5 // false

Greater than or equal to (>=): Trả về true nếu toán hạng thứ nhất lớn hơn hoặc bằng toán hạng thứ hai.

5 >= 5 // true 6 >= 2 // true 3 >= 4 // false

Less than or equal to (<=): Trả về true nếu toán hạng thứ nhất nhỏ hơn hoặc bằng toán hạng thứ hai.

4 <= 4 // true 2 <= 3 // true 5 <= 4 // false

***Logical Operators (Toán tử logic)***

Toán tử logic được sử dụng để kết hợp nhiều điều kiện hoặc giá trị boolean.

AND (&&): Trả về true nếu cả hai toán hạng đều đúng.

true && true // true true && false // false

OR (||): Trả về true nếu bất kỳ toán hạng nào là đúng.

true || false // true false || false // false

NOT (!): Trả về true nếu toán hạng sai và ngược lại.

!true // false !false // true

**Arithmetic Operators in C++**

***1. Addition Operator (+)***

Nó cộng hai số lại với nhau

int sum = a + b;

1. ***Subtraction Operator (-)***

Nó trừ một số này cho một số khác.

int difference = a - b;

1. ***Multiplication Operator (\*)***

Nó nhân hai số với nhau.

int product = a\*b;

1. ***Division Operator (/)***

Phép chia này chia một số cho một số khác. Lưu ý rằng nếu cả hai toán hạng đều là số nguyên, phép chia sẽ thực hiện chia số nguyên và kết quả sẽ là một số nguyên.

int quotient = a / b; // integer division

float quotient = float(a) / float(b); // floating-point division

1. ***Modulus Operator (%)***

Tính toán phần dư của phép chia số nguyên.

int remainder = a % b;

1. ***Increment Operator (++)***

Toán tử này tăng giá trị của một biến lên 1. Có hai cách để sử dụng toán tử này: tiền tố (++x) và hậu tố (x++). Tiền tố tăng giá trị trước khi trả về, trong khi hậu tố trả về giá trị trước rồi mới tăng giá trị.

int x = 5;

int y = ++x; // x = 6, y = 6

int z = x++; // x = 7, z = 6

1. ***Decrement Operator (--)***

Nó giảm giá trị của một biến đi 1. Nó cũng có thể được sử dụng ở dạng tiền tố (--x) và hậu tố (x--)

int x = 5;

int y = --x; // x = 4, y = 4

int z = x--; // x = 3, z = 4

Đây là các toán tử số học cơ bản trong C++ cho phép bạn thực hiện các phép toán trên các biến. Hãy sử dụng chúng kết hợp với các cấu trúc điều khiển khác, chẳng hạn như vòng lặp và câu lệnh điều kiện, để xây dựng các chương trình phức tạp hơn.

**Logical Operators in C++**

Toán tử logic được sử dụng để thực hiện các phép toán logic trên các biểu thức đã cho, chủ yếu để kiểm tra mối quan hệ giữa các biến hoặc giá trị khác nhau. Chúng trả về một giá trị boolean, tức là đúng (1) hoặc sai (0) dựa trên kết quả đánh giá.

C++ cung cấp các toán tử logic sau:

- ***Toán tử AND (&&)*** : Toán tử AND kiểm tra xem cả hai toán hạng/điều kiện có đúng hay không, thì biểu thức sẽ đúng. Nếu bất kỳ điều kiện nào sai, toàn bộ biểu thức sẽ sai.

(expression1 && expression2)

Example:

int a = 5, b = 10;

if (a > 0 && b > 0) {

std::cout << "Both values are positive.\n";

}

-***Toán tử OR (||)*** : Toán tử OR kiểm tra xem một trong hai toán hạng/điều kiện có đúng hay không, thì biểu thức sẽ đúng. Nếu cả hai điều kiện đều sai, biểu thức sẽ trả về giá trị false.

(expression1 || expression2)

Example:

int a = 5, b = -10;

if (a > 0 || b > 0) {

std::cout << "At least one value is positive.\n";

}

- ***Toán tử NOT (!) :*** Toán tử NOT đảo ngược kết quả của điều kiện/biểu thức mà nó được áp dụng. Nếu điều kiện là đúng, toán tử NOT sẽ làm cho nó sai và ngược lại.

!(expression)

Example:

int a = 5;

if (!(a < 0)) {

std::cout << "The value is not negative.\n";

}

Sử dụng các toán tử này, bạn có thể tạo các biểu thức logic phức tạp hơn, ví dụ:

int a = 5, b = -10, c = 15;

if (a > 0 && (b > 0 || c > 0)) {

std::cout << "At least two values are positive.\n";

}

Phần này bao gồm những thông tin cần thiết về toán tử logic trong C++.