

# Arithmetic Operators

## Example

```
int x = 10;
int y = 3;
cout << (x + y) << "\n"; // 13
cout << (x - y) << "\n"; // 7
cout << (x * y) << "\n"; // 30
cout << (x / y) << "\n"; // 3 (integer division)
cout << (x % y) << "\n"; // 1
```

When dividing two integers in C++, the result will also be an integer. For example, **10 / 3** gives **3**. If you want a decimal result, use **float** or **double** values, like **10.0 / 3**.

## **z++** → **Post-increment**

- Increments the variable **after** its current value is used.
- The expression returns the **old value**, then adds 1.

```
int z = 5;
int x = z++; // post-increment
```

x gets **z's old value** → x = 5

Then  $z$  increases by 1  $\rightarrow z = 6$

**Result:**

$x = 5, z = 6$

## **$++z \rightarrow$ Pre-increment**

- Increments the variable **before** its value is used.
- The expression returns the **new value** (after adding 1).

```
int z = 5;
```

```
int x = ++z; // pre-increment
```

$z$  increases by 1  $\rightarrow z = 6$

$x$  gets the **new value**  $\rightarrow x = 6$

**Result:**

$x = 6, z = 6$

# Operations Hierarchy Example

---

$3 * (6 + 2) / 12 - (7 - 5) ^ 2 * 3$	( ) first
$= 3 * 8 / 12 - 2 ^ 2 * 3$	^ next
$= 3 * 8 / 12 - 4 * 3$	Mult / Div (L to R)
$= 24 / 12 - 4 * 3$	Mult / Div (L to R)
$= 2 - 12$	Add / Subtr
$= -10$	

## Example 1

```
int result1 = 2 + 3 * 4;
int result2 = (2 + 3) * 4;

cout << result1 << "\n";
cout << result2 << "\n";
```

## Example 2

```
int result1 = 10 - 2 + 5;
int result2 = 10 - (2 + 5);

cout << result1 << "\n";
cout << result2 << "\n";
```

## Assignment Operators

```
int x = 3;
```

```
x += 3; // x = x + 3
```

```
x -= 3; // x = x - 3
```

```
x *= 2; // x = x * 2
```

```
x /= 4; // x = x / 4
```

```
x %= 3; // x = x % 3
```

## Comparison Operators

Comparison operators are used to compare two values (or variables). This is important in programming, because it helps us to find answers and make decisions.

The return value of a comparison is either **1** or **0**, which means **true** (1) or **false** (0). These values are known as **Boolean values**

Example:

```
int age = 18;
```

```
cout << (age >= 18) << "\n";
```

```
cout << (age < 18) << "\n";
```

example:

```
int passwordLength = 5;

cout << (passwordLength >= 8) << "\n";
cout << (passwordLength < 8) << "\n";
```

## Logical Operators

As with [comparison operators](#), you can also test for **true** (1) or **false** (0) values with **logical operators**.

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	<code>x &lt; 5 &amp;&amp; x &lt; 10</code>
	Logical or	Returns true if one of the statements is true	<code>x &lt; 5    x &lt; 4</code>
!	Logical not	Reverse the result, returns false if the result is true	<code>!(x &lt; 5 &amp;&amp; x &lt; 10)</code>