

## 1. What is Arithmetic Operators

\* Arithmetic operator is used to perform mathematical operations such as addition, subtraction, multiplication, division, modulus, etc. on the given operands.

Example :  $5 + 3 = 8$  // addition or unary plus  
 $5 - 3 = 2$  // subtraction or unary minus  
 $2 \cdot 4 = 8$  // multiplication  
 $8 / 2 = 4$  // division

2. In programming (Java, C, C++, JavaScript etc.), the increment operator (++) increases the value of a variable by 1. Similarly, the decrement operator (--) decreases the value of a variable by 1.

Example :  $a = 5$  ++a // a becomes 6  
 ++a // a becomes 7  
 --a // a becomes 6  
 --a // a becomes 5.



Mo Tu We Th Fr Sa Su

Memo No. \_\_\_\_\_

Date / /

## 2. Assignment operator

\* assigns the value of its right-hand operand to a variable, a property, or an indexed element given by its left-hand operand

Operator	Example	Equivalent Expression
$\% =$	allowance $\% = 1000$	allowance = allowance $\% 1000$

$\ll =$	result $\ll = \text{num}$	result <del><math>\ll</math></del> = result $\ll \text{num}$
---------	---------------------------	--

$\gg =$	form $\gg = 1$	form = $\gg 1$
---------	----------------	----------------

$\& =$	mark $\& = 2$	mark = mark $\& 2$
--------	---------------	--------------------

## 4. Relational operators

\* checks the relationship between two operands. If the relation is true, it returns 1; if the relation is false, it returns value 0. Relational operators are used in decision making and loops.





Mo Tu We Th Fr Sa Su

Memo No. \_\_\_\_\_

Date      /      /

Operator	Meaning
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
==	equality (equal to)
!= or <>	inequality (not equal to)

## 5. Logical operator

\* to perform various logical operation on any set of given expressions. The logical operators in C are used for combining multiple constraints/conditions or for complementing the evaluation of any original condition that is under consideration.

Operator	Name	Example
&&	AND. true only if both operands are true	<code>x=5; (x&gt;1) &amp;&amp; (x&lt;4)</code>
	OR. true either operand is true	<code>x=5; (x&gt;1)    (x&lt;4)</code>
~	NOT. Changes true to false and false to true	<code>a="HELLO"; ~isalpha(a)</code>

## 6. Conditional Operators (?:)

\* A ternary operator (it takes three operands).

The conditional operator works as follows.

The first operand is implicitly converted to bool.

It is evaluated and all side effects are completed before continuing.

Example:

```
int main ()
```

```
{
```

```
    int i=1, j=2;
```

```
    cout << (i > j ? i : j) << " is greater." << endl;
```

```
}
```