prefix

Postfix
( ) ++

++ (Increment)

Increases the value of a variable by 1 ++x

```
#include <iostream>
using namespace std;
int main() {
  int x = 5;
  ++x;
  cout << x;
  return 0;
}
```

6

-- (Decrement)

Decreases the value of a variable by 1 --x

```
#include <iostream>
  using namespace std;
  int main() {
    int x = 5;
    --x;
    cout << x ;
  return 0;
  }
```

4

C++ Assignment Operators

Assignment operator (=)

These operators assign values to variables.

```
#include <iostream>
using namespace std;
int main() {
  int x = 10;
  cout << x;
  return 0;
}
```

10

Addition assignment operator (+=)

↳ (+=) adds a value to a variable.

```
#include <iostream>
using namespace std;
int main() {
  int x = 25;
  x += 5;
  cout << x;
  return 0;
}
```

30

All assignment operators:

Operator	Example	Same As
=	x = 10	x = 10
+=	x += 2	x = x + 2
-=	x -= 2	x = x - 2
*= (multiplication assignment)	x *= 2	x = x * 2
/= (division assignment)	x /= 2	x = x / 2
%= (integer remainder, modulo assignment)	x %= 2	x = x % 2
&= (and assignment)	x &= 2	x = x & 2
|= (or assignment)	x |= 2	x = x | 2
^= (xor assignment)	x ^= 2	x = x ^ 2

Bitwise operators

Example: /= (for x = 5)

```
#include <iostream>
using namespace std;
int main(){
    double x = 5;
    x /= 2;
    cout << x;
    return 0;
}
```

2.5

Ex: remainder x=10, dusion for 3

```
#include <iostream>
using namespace std;
int main( ){
  int x = 10;
  x %= 3;
  cout << x;
  return 0;
}
```

1

Ex: x = 5 division = 3

```
#include <iostream>
using namespace std;
int main() {
  double x = 5;
  x /= 3;
  cout << x;
  return 0;
}
```

1.66667

C++ Comparison Operators → They are comparing two values.

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y (x ≥ y)
<=	Less than or equal to	x <= y (x ≤ y)

```
#include <iostream>
using namespace std;

int main() {
  int x = 5;
  int y = 3;
  cout << (x == y); //returns 0 (false) because 5 is not
                    equal to 3
  return 0;
}
```

0

```
#include <iostream>
using namespace std;
int main() {
  int x = 5;
  int y = 3;
  cout << (x >= y); //returns 1 (true) because five
                    is greater than, or equal, to 3
  return 0;
}
```

1

C++ Logical Operators

These operators are used to determine the logic between variables or values.

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	x < 5 && x < 10
||	Logical or	Returns true if one of the statements is true	x < 5 || x < 4
!	Logical not	Reverses the result, returns false if the result is true	! (x < 5 && x < 10)

```
#include <iostream>
using namespace std;
int main() {
  int x = 5;
  [redacted]
  cout << (x > 3 && x<10); //returns true (1)
    because 5 is greater than 3 AND 5 is less than 10
  return 0;
}
```

1