Using Short hand & Increment / Decrement op

```
int main ( )
{
        int i;
        i = 1;
        while (i <= 10)
       {
          printf("\n %d", i);
          i = i + 1;
       }
 return 0;
}
```

$$\begin{tikzcd}[row sep=small]
 & i += 1; \\
\boxed{i = i + 1;} \arrow[ur] \arrow[r] \arrow[dr] & i++; \\
 & ++i;
\end{tikzcd}$$

① $\boxed{a = a + b;}$

or

$\underline{a \mathrel{+=} b;}$

② $i = i * j;$

or

$i \mathrel{*=} j;$

③ $x = x - y;$

or

$x \mathrel{-=} y;$

$$\begin{tikzcd}
 & \text{Short hand ops} \\
\boxed{\begin{array}{l} += \\ -= \\ *= \\ /= \\ \%= \end{array}} \arrow[ur] \arrow[r] \arrow[dr] & \text{Compound op} \\
 & \text{Arithmetic Assignment Operator}
\end{tikzcd}$$

```
int a;

a = 10;
```

$$\begin{tikzcd}
 & \texttt{a++;} \\
\boxed{\texttt{a = a + 1;}} \arrow[ur] \arrow[r] & \texttt{++a;}
\end{tikzcd}$$

```
printf("%d", a);
```

Using Increment Op

++

$$\begin{tikzcd}
 & \text{Unary Incr Op} \arrow[dl] \arrow[dr] & \\
\underline{\text{Post Incr}} & & \underline{\text{Pre Incr}} \\
\text{a++;} & & \text{++a;}
\end{tikzcd}$$

```
int   b=10;

b++;

printf("%d", b);
       ↳ 11
```

```
int  b=10;

++b;

printf("%d", b);
       ↳ 11
```

```
int a, b=10;
```

$$\begin{tikzcd}
 & a = b; \\
\boxed{a = b{+}{+};} \arrow[ur] \arrow[dr] & \\
 & b{+}{+};
\end{tikzcd}$$

```
printf("%d %d", a,b);
        10  11
```

```
int a, b=10;
```

$$\begin{tikzcd}
 & {+}{+}b; \\
\boxed{a = {+}{+}b;} \arrow[ur] \arrow[dr] & \\
 & a = b;
\end{tikzcd}$$

```
printf("%d %d", a,b);
        11  11
```

```
int x=2, y, z;
```

$\boxed{\cancel{2}\,\cancel{3}\,4}$ $\boxed{2}$ $\boxed{\cancel{3}\,4}$

x y z

$$\begin{tikzcd}
 & y = x; \\
\boxed{y = x\underline{{+}{+}};} \arrow[ur] \arrow[dr] & \\
 & x{+}{+};
\end{tikzcd}$$

$$\begin{tikzcd}
 & {+}{+}x \\
\boxed{z = \underline{{+}{+}}x;} \arrow[ur] \arrow[dr] & \\
 & z = x
\end{tikzcd}$$

```
printf("%d %d %d", x,y,z);
```

$\cancel{2}$ 2 3 | ~~2 3 3~~ | ~~2 3 4~~

~~3 2 4~~ | ~~3 3 4~~ | ~~2 3 2~~

~~4 2 3~~ | ~~3 2 4~~

✓ $\boxed{4 \quad 2 \quad 4}$

int x = 2, y, z;

~~2~~ ~~3~~ 4 | ~~?~~ 3 | ~~?~~ 3
x | y | z

$$\begin{tikzcd} \boxed{y = ++x;} \arrow[r] \arrow[dr] & ++x; \\ & y = x; \end{tikzcd}$$

$$\begin{tikzcd} \boxed{z = x++;} \arrow[r] \arrow[dr] & z = x; \\ & x++; \end{tikzcd}$$

```
printf("%d %d %d", x, y, z);
```

✓ $\boxed{4 \quad 3 \quad 3}$

① Pre

② Arithmetic

③ Assignment

④ Post

```
int a=5, b=10, c;

c = a++ + b++;

printf("%d %d %d", a, b, c);
        6  11  15
```

```
int a=5, b=10,c;

c= ++a + b++;

printf("%d %d %d", a, b, c);
        6  11  16
```

Whenever we use incr/decr operator multiple times on the SAME VAR in the SAME EXPR then the behaviour becomes totally COMPILER DEPENDENT

```
int a=10, b;
                              TC

b= a++ + a++;

printf("%d %d", a, b);
        12  20
```

```
int a=10, b;
                                    TC
b=   a++   +  ++a;

printf("%d %d", a, b);
         12    22
```

In MingW

① L -> R

② Post : Use and then incr

③ Pre : Keep on incr & after exp ends
then use the value

12
| ~~10 11~~ | a | ? | b

```
int   a=10, b;
```

~~Mingw~~

```
b= a++  +  a++;
   10   +  11

printf("%d %d", a, b);
        12  21
```

```
int  a=10, b;
```

| ~~10 11~~ 12 | a

```
b=   ++a   +   a++;
     12?   +   11

printf("%d %d", a, b);
        12  23
```

++ V/s	+
① Incr Op	① Arithmetic Op
② Unary Op	② Binary Op
③ Always changes the value of its opnd	③ Never changes the value of its opnd.

Types of Operator (Based Upon No of operands)

$$\begin{tikzcd}
 & \text{Types of Operator} \arrow[dl, no head] \arrow[d, no head] \arrow[dr, no head] & \\
\text{Unary} & \text{Binary} & \text{Ternary}
\end{tikzcd}$$

Unary
(Req only 1 opnd)
① a++;
② ++a;

Binary
(Req 2 opnd)
① x+y

Ternary
(Req 3 opnd)
① (test_cond) ? (st) : (st);

Operator	Name	Category
① =	Assignment	Binary
② = =	Equality Comps	Binary
③ &	Addr of op	Unary
④ +	Arithmetic +	Binary
⑤ −	Unary − OR Negation op	Unary

```
int  a;
a = -6;
```

```
int  a;
a =  7 - 6;
        ↓
     7 + (-6)
```

① int a=10, b;

$$\begin{tikzcd} b = \arrow[d] & \boxed{a++;} \arrow[d] \\ 10 & 11 \end{tikzcd}$$

③ int a=10, b;

$$\begin{tikzcd} b = \arrow[d] & a+1; \arrow[d] \\ 11 & 10 \end{tikzcd}$$

② int a=10, b;

$$\begin{tikzcd} b = \arrow[d] & \boxed{++a;} \arrow[d] \\ 11 & 11 \end{tikzcd}$$

$$\begin{tikzcd} & \underline{\text{Unary Decr Op}} \arrow[d] & \\ & -- \arrow[dl] \arrow[dr] & \\ \underline{\text{Pre Decr}} & & \underline{\text{Post Decr}} \end{tikzcd}$$

--i;

i--;

```
a++;

X a++++;

X ++a++;
```