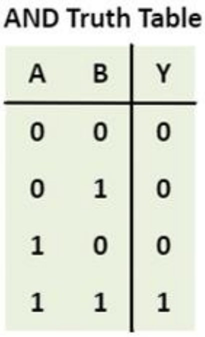
What is operator?

Operator is a symbol that is used to perform operations according to user requirement.

Types :-

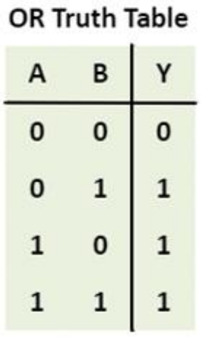
1. Arithmetic – (+ , - , * , / , %)
2. Relational – (< , > , >= , <= , != , ==)
3. Logical – (&& , || , !)
4. Increment & Decrement (Pre & Post)
5. Assignment – (Simple[=], Compound[+=, -=, etc])
6. Bitwise (&, |, >>, <<, ^, ~)
7. Ternary Operator(?:)

Bitwise (Binary Concept)



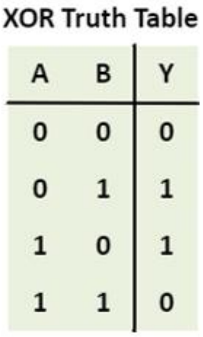
AND Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



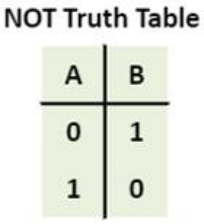
OR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



XOR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



NOT Truth Table

A	B
0	1
1	0

Bitwise AND (&)

Bitwise AND operation is performed between two integers, It will compare each bit on the same position and the result bit will be set(1) only and only if both corresponding bits are set(1). The symbol which is used to perform bitwise AND operation is &.

1 1 1

1 0 0 &

----

1 0 0

Bitwise OR (|)

If Bitwise OR operation is performed between two integers, It will compare each bit on same position and the result bit will be set(1) if any of corresponding bits are set(1). The symbol which is used to perform bitwise OR operation is |.

```
1 1 1

1 0 0 |

----------

1 1 1
```

Bitwise XOR (^)

If Bitwise XOR operation is performed between two integers , It will compare each bit on same position and the result bit will be set(1) if any of corresponding bits differ i.e. one of them should be 1 and other should be zero. The symbol which is used to perform bitwise XOR operation is ^.

```
1 1 1

1 0 0 ^

----------

0 1 1
```

Bitwise NOT (~)

The Bitwise NOT operation is performed on a single number. It change the current bit to it’s complement , i.e. if current bit is 0 then in result it will be 1 and if current bit is 1 then it will become 0. It is denoted by the symbol ~.

```
1 0 1 ~

----------

0 1 0
```

Left Shift (<<)

This operator shifts the bits of Integer to left side by specific number (As mentioned). This left shift operation is equivalent to multiplying the integer by 2 power number of positions shifted. The symbol which is used to represent Left Shift Operator is <<.

Consider we have an integer 5, and we will left-shift its bits by 2 positions. The operation will be represented as x << 2.
The number 5 is represented as 101 in binary. We will add some zeros at the beginning to left shift the bits. Therefore it will be represented as 00000101. Now, we will move all the bits two positions to left and we will fill the empty positions with 0. Therefore it will become 00010100 which is 20. As mentioned earlier left shifting the number by two bits means multiplying it by 2 raised to 2 which is 4. 5*4 = 20 shows the above mentioned statement.

Right Shift (>>)

This operator shifts the bits of Integer to right side by specific number (As mentioned). This right shift operation is equivalent to dividing the integer by 2 power number of positions shifted. The symbol which is used to represent Left Shift Operator is >>.

Consider we have an integer 16, and we will right-shift its bits by 2 positions. The operation will be represented as x >> 2.
The number 16 is represented as 10000 in binary. We will add some zeros at the beginning to right shift the bits. Therefore it will be represented as 00010000. Now, we will move all the bits two positions to right and we will fill the empty positions with 0. Therefore it will become 00000100 which is 4. As mentioned earlier right shifting the number by two bits means dividing it by 2 raised to 2 which is 4. 16*4 = 4 shows the above mentioned statement.