**Bitwise Operators**

Bitwise operators are used to perform operations on **binary representations** of integers. These operators work at the **bit level**, meaning they directly manipulate individual bits of numbers.

| **Operator** | **Name** | **Symbol** | **Description** |
| --- | --- | --- | --- |
| AND | Bitwise AND | & | Sets each bit to 1 if both bits are 1 |
| OR | Bitwise OR | | | Sets each bit to 1 if one of the bits is 1 |
| XOR | Bitwise XOR | ^ | Sets each bit to 1 if only one of the bits is 1 |
| NOT | Bitwise NOT | ~ | Inverts all the bits (1 becomes 0, and 0 becomes 1) |
| << | Left Shift | << | Shifts bits to the left, filling with 0s on the right a=5 00001010 a<<1 |
| >> | Right Shift | >> | Shifts bits to the right, discarding rightmost bits |

**Example (Using 8-bit representation)**

Let’s use a = 5 and b = 3.

a = 5 -> 00000101

b = 3 -> 00000011

* a & b → 00000001 → **1**
* a | b → 00000111 → **7**
* a ^ b → 00000110 → **6**
* ~a → 11111010 → **-6** (in two's complement form)
* a << 1 → 00001010 → **10**
* a >> 1 → 00000010 → **2**