BITWISE OPERATORS:

Definition: In computer programming, a bitwise operation operates on one or more-bit patterns or binary numerals at the level of their individual bits. It is a fast and simple action, directly supported by the processor, and is used to manipulate values for comparisons and calculations.

These are of 6 types:

|  |  |
| --- | --- |
| Syntax | Operator |
| & | Bitwise AND |
| | | Bitwise OR |
| ^ | Bitwise Ex-OR |
| ~ | 1’s complement (NOT) |
| << | Left shift |
| >> | Right shift |

For example: Consider a=10 and b=5,

Bitwise AND: a&b gives 0

Bitwise OR: a|b gives 15

Bitwise Ex-OR: a^b gives 15

1’s complement: ~a gives some negative value

Left shift: a<<1 gives 20

Right shift: a>>1 gives 5

TERNARY OPERATORS:

Definition: In C Programming language, ternary operator allows executing different code depending on the value of a condition, and the result of the expression is the returned value of the executed code. The main advantage of using ternary operator is to reduce the number of line codes and improve the performance of application.

Syntax: expression-1? expression-2: expression-3

Example: a=2, b=3.

(a>b)? PRINTF(“b is less”); : printf(“a is less”);

It prints: “a is less” without quotes.