(5)

Bitwise operators :-

Bitwise operators perform manipulations of data at bit level. These operators also perform shifting of bits from right to left.

Bitwise operators are not applied to float or double

operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
<<	left shift
>>	right shift

Shift operators → { << , >> }

Truth Table for bitwise &, | and ^.

a	b	a&b	a|b	a^b
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

a&b : 1×1

a|b : $1+1 = 1\underline{0}$ excluded

a^b : $1+1 = 1\underline{0}$ excluded value is taken

─ The bitwise shift operator shifts the bit value.

─ The left operand specifies = The value in binary to be shifted

Right operand specifies = No. of position to be shifted

Syntax :-   operand 1   operator   operator 2

e.g :-   a & b

program :-

```c
#include <stdio.h>
int main ()
{
    int  a=0, b=1, ans ;
    ans =    a & b ;
    printf ( " %d ", ans );
    return 0 ;
}
```

output :-   0

Ternary Operator

It is a Special type of decision making operator.
It is actually the if condition that we use in C language
decision making, but using conditional operator,
we turn the if condition Statement into a short and
Simple operator.

Syntax :- Boolean Expression ? First Statement : Second Statement

OR

Syntax :- Condition ? Value_if_true : Value_if_false

Example :-
```c
#include <stdio.h>
int main ()
{
    int a = 50, b = 40, ans ;
    ans = ( a > b ? 1 : 0 );
    printf ( "%d", ans ) ;
    return 0 ;
}
```

output :-   1