025 -
BITWISE OPERATORS IN C

The bitwise operators are the operators used to perform the operations on the data at the bit-level.

When we perform the bitwise operations, then it is also known as bit-level programming.

It consists of two digits, either 0 or 1. It is mainly used in numerical computations to make the calculations faster.

BITWISE " AND " OPERATOR (&)

If the corresponding bits of both the operands are 1, then the output of the bitwise AND operation is 1; otherwise, the output would be 0.

BITWISE " OR " OPERATOR (|)

If the bit value of any of the operand is 1, then the output would be 1, otherwise 0.

BITWISE " EXCLUSIVE OR " OPERATOR (^)

If the corresponding bit of any of the operand is 1 then the output would be 1, otherwise 0.

BITWISE " COMPLEMENT " OPERATOR (~)

The bitwise complement operator is also known as one's complement operator.

It takes only one operand or variable and performs complement operation on an operand.

When we apply the complement operation on any bits, then 0 becomes 1 and 1 becomes 0.

BITWISE " LEFT-SHIFT " OPERATOR (<<)

It is an operator that shifts the number of bits to the left-side.

```
Bitwise left-shift operators syntax :-

Operand << n ;
```

Where, Operand is an integer expression on which we apply the left-shift operation. n is the number of bits to be shifted.

BITWISE " RIGHT-SHIFT " OPERATOR (>>)

It is an operator that shifts the number of bits to the right side.

```
Bitwise left-shift operators syntax :-

Operand >> n ;
```

Where, Operand is an integer expression on which we apply the right-shift operation. n is the number of bits to be shifted.
