**Operator Precedence & Associativity**

The concept of operator precedence and associativity in C helps in determining which operators will be given priority when there are multiple operators in the expression. It is very common to have multiple operators in C language and the compiler first evaluates the operator with higher precedence. It helps to maintain the ambiguity of the expression and helps us in avoiding unnecessary use of parenthesis.

**Easy Trick to Remember the Operators Precedence:**

**PEMDAS (Parentheses, Exponents, Multiplication and Division (from left to right), Addition and Subtraction (from left to right))**: This mnemonic is a common starting point for remembering operator precedence in many languages, including C. Operators within the same group have the same precedence, and groups are evaluated from left to right.

**Associativity:**

Operator associativity is used when two operators of the same precedence appear in an expression. Associativity can be either from Left to Right or Right to Left.

1. **Left Associative Operators**: Operators that are left associative are evaluated from left to right. This means that if there are multiple operators of the same precedence level in an expression, they are evaluated from the leftmost to the rightmost.

Examples of left associative operators in C include:

* + Arithmetic operators (+, -, \*, /, %,)
  + Bitwise operators (&, |, ^, <<, >>)
  + Relational operators (<, <=, >, >=, ==, !=)
  + Logical operators (&&, ||)
  + Others( Comma(,), \*\*,%)

For example:

int result = 10 - 5 + 3; // Evaluates as (10 - 5) + 3 = 8

1. **Right Associative Operators**: Operators that are right associative are evaluated from right to left. This means that if there are multiple operators of the same precedence level in an expression, they are evaluated from the rightmost to the leftmost.

There are a few operators that are right-to-left associative. The most important one to remember is the assignment operator (=) and its compound variations

(++, --, =, +=, -=, \*=, /=, %=, &=, |=, ^=, <<=, >>=etc.),

(ternary conditional expression (condition ? expression\_if\_true : expression\_if\_false)),

Pre Increment , Post Increment , Pre Decrement, Post Discernment

For example (hypothetical):

int result = 2 ^ 3 ^ 2; // Evaluates as 2 ^ (3 ^ 2) = 2 ^ 9 = 511 (if ^ was right associative)

In C, most operators are left associative by default. Right associative operators are rare in C, with assignment (=) being one of the few examples.