Operators in C

An operator is simply a symbol that is used to perform operations.

or

An **operator** is a symbol in C programming that performs operations on one or more **operands**. Operands are the values or variables on which the operator acts. Operators are essential for creating expressions and performing calculations, comparisons, and other operations.

For example:
In the expression a + b,

- + is the **operator**.
- a and b are the **operands**.

Types:

There are following types of operators to perform different types of operations in C language.

1. Arithmetic Operators:

These operators perform basic mathematical operations.

Operator	Description	Example
+	Addition	a + b adds a and b.
-	Subtraction	a - b subtracts b from a.
*	Multiplication	a * b multiplies a and b.
/	Division	a / b divides a by b.
%	Modulus (remainder)	a % b gives remainder of a divided by b.

Example:

```
#include <stdio.h>

int main() {
    int a = 10, b = 3;
    printf("Addition: %d\n", a + b);  // 13
    printf("Division: %d\n", a / b); // 3
    printf("Modulus: %d\n", a % b);  // 1
    return 0;
}
```

2. Relational (Comparison) Operators

These operators compare two operands and return a boolean result (true or false).

Operator	Description	Example
==	Equal to	a == b (true if a equals b).
!=	Not equal to	a != b (true if a is not equal to b).
>	Greater than	a > b (true if a is greater than b).
<	Less than	a < b (true if a is less than b).
>=	Greater than or equal to	a >= b (true if a is greater than or equal to b).
<=	Less than or equal to	a <= b (true if a is less than or equal to b).

Example:

```
#include <stdio.h>
int main() {
   int a = 5, b = 10;
   printf("Is a less than b? %d\n", a < b);  // 1 (true)
   return 0;
}
```

---

3. Logical Operators

These operators are used to combine multiple conditions.

Operator	Description	Example
&&	Logical AND	a && b (true if both a and b are true).
`	`	
!	Logical NOT	!a (true if a is false).

Example:

```
#include <stdio.h>
int main() {
   int a = 1, b = 0;
   printf("Logical AND: %d\n", a && b);  // 0 (false)
   printf("Logical OR: %d\n", a || b);  // 1 (true)
```

```
    return 0;
}
```

---

4. Bitwise Operators

These operators perform bit-level operations.

Operator	Description	Example
&	Bitwise AND	a & b (performs AND on bits).
`	`	Bitwise OR
^	Bitwise XOR	a ^ b (performs XOR on bits).
~	Bitwise Complement	~a (flips bits of a).
<<	Left Shift	a << 2 (shifts bits of a left).
>>	Right Shift	a >> 2 (shifts bits of a right).

---

5. Assignment Operators

These are used to assign values to variables.

Operator	Description	Example
=	Assignment	a = b (assigns value of b to a).
+=	Add and assign	a += b (equivalent to a = a + b).
-=	Subtract and assign	a -= b (equivalent to a = a - b).
*=	Multiply and assign	a *= b (equivalent to a = a * b).
/=	Divide and assign	a /= b (equivalent to a = a / b).

Example:

```
#include <stdio.h>
int main() {
    int a = 5;
    a += 10;  // a = a + 10
    printf("Value of a: %d\n", a);  // 15
    return 0;
}
```

---

6. Increment and Decrement Operators

These are used to increase or decrease the value of a variable by 1.

Operator	Description	Example
++	Increment	a++ or ++a.
--	Decrement	a-- or --a.

Example:

```
#include <stdio.h>
int main() {
   int a = 5;
   printf("Post-increment: %d\n", a++);  // 5
   printf("Value of a: %d\n", a);       // 6
   return 0;
}
```

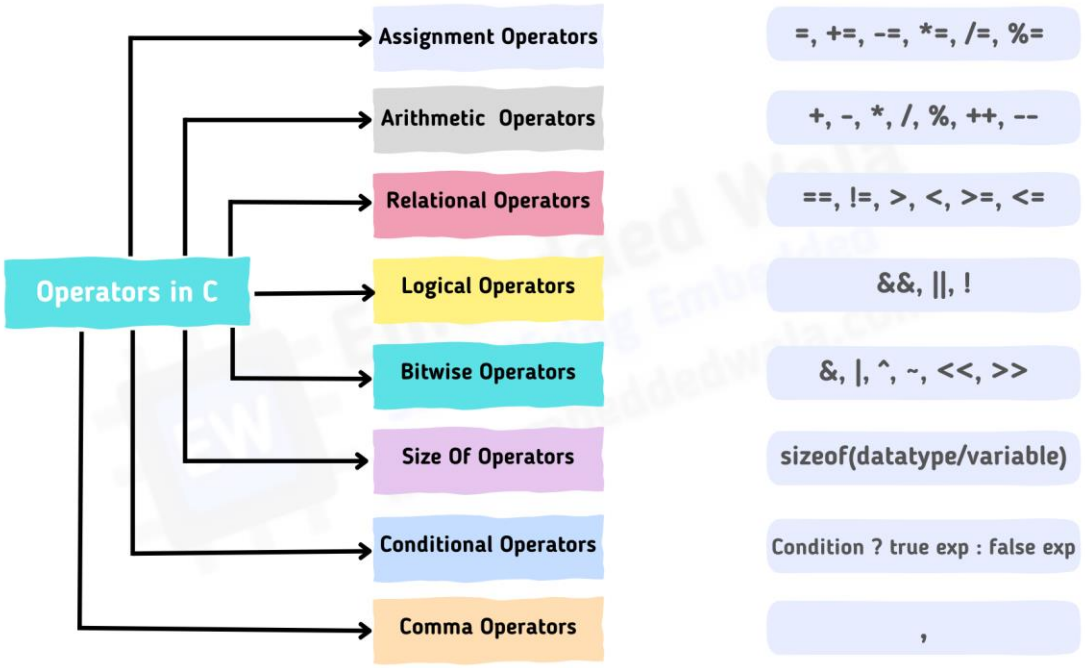
---

7. Special Operators

- **Conditional (Ternary) Operator**: condition ? expression1 : expression2
    - Example: a > b ? a : b (returns the greater of a or b).
- **Comma Operator**: , (Evaluates multiple expressions, returns the last value).
    - Example: a = (b = 3, b + 2) (assigns 5 to a).
- **Sizeof Operator**: sizeof (Returns the size of a variable or type).
    - Example: sizeof(int).

Example:

```
#include <stdio.h>
int main() {
   int a = 5, b = 10;
   printf("Greater number: %d\n", a > b ? a : b);  // 10
   printf("Size of int: %zu\n", sizeof(int));      // 4 (on most systems)
   return 0;
}
```


Operators in C
Assignment Operators
=, +=, -=, *=, /=, %=
Arithmetic Operators
+, -, *, /, %, ++, --
Relational Operators
==, !=, >, <, >=, <=
Logical Operators
&&, ||, !
Bitwise Operators
&, |, ^, ~, <<, >>
Size Of Operators
sizeof(datatype/variable)
Conditional Operators
Condition ? true exp : false exp
Comma Operators
,