

# Operators in C program

5. Write notes explaining each type of operator in C: arithmetic, relational, logical, assignment, increment/decrement, bitwise, and conditional operators.

→ Types of Operators in C

→ **Arithmetic Operators:** Used for basic mathematical operations.

→ Examples: + (addition), - (subtraction), \* (multiplication), / (division), % (modulus).

→ Example: `a + b` adds `a` and `b`.

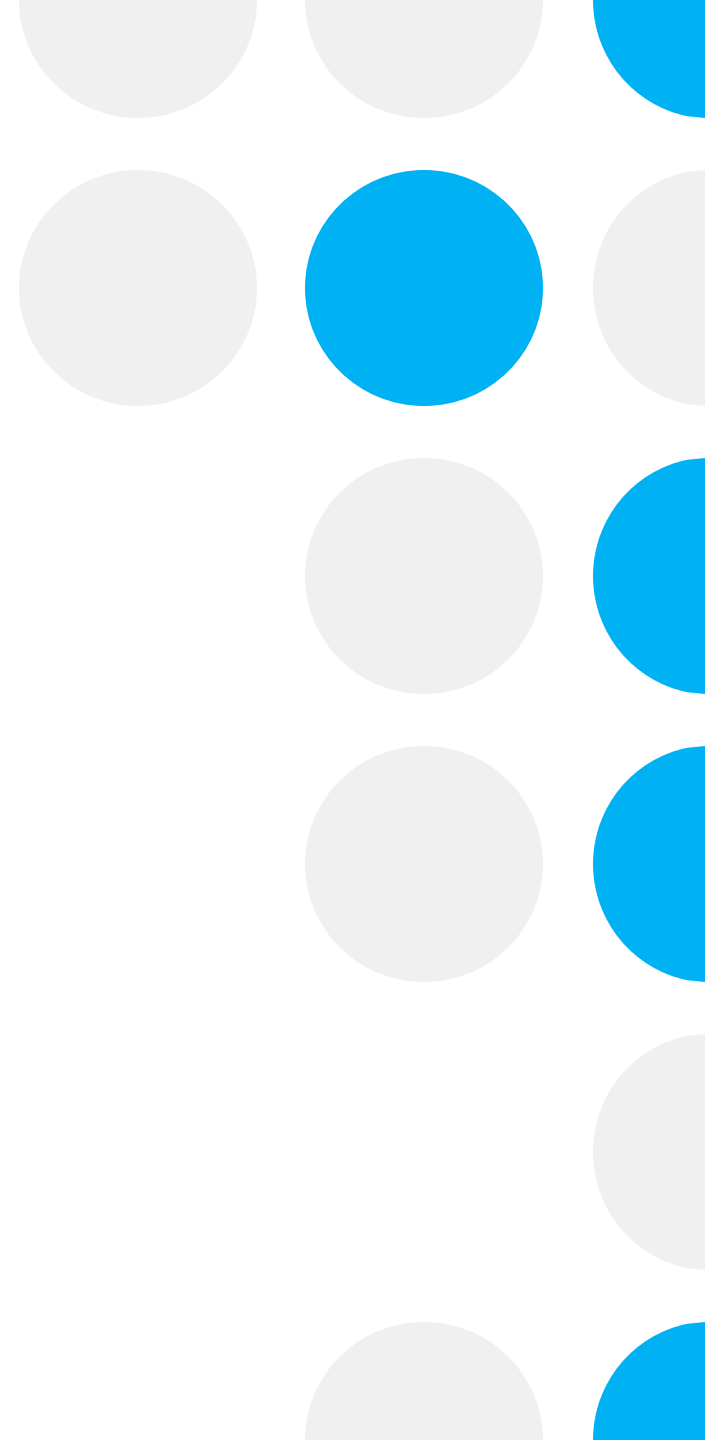
→ **Relational Operators:** Compare values and return true (1) or false (0). → Examples:

→ `==` (equal to), `!=` (not equal to), `>` (greater than), `<` (less than), `>=` (greater than or equal to), `<=` (less than or equal to).

→ Example: `a < b` checks if `a` is less than `b`.

→ **Logical Operators:** Used for logical operations, typically with conditional statements.

---



→Examples:

→&& (logical AND), || (logical OR), ! (logical NOT). o Example: (a > 5 && b < 10) is true if both conditions are true.

→ Assignment Operators: Assign values to variables.

→ Examples:

→ = (simple assignment), += (add and assign), -= (subtract and assign), \*= (multiply and assign), /= (divide and assign). o Example: a += 5 increases a by 5.

→ Increment/Decrement Operators: Increase or decrease variable values by 1.

→Examples:

→++ (increment), -- (decrement). o Example: a++ increases a by 1.

→ Bitwise Operators: Perform bit-level operations.

→Examples:

→& (AND), | (OR), ^ (XOR), ~ (NOT), << (left shift), >> (right shift).

→Example:

→a & b performs bitwise AND on a and b.

→ Conditional (Ternary) Operator: A shorthand for if-else statements. O

→Syntax: condition ? expr1 : expr2.

→Example: (a > b) ? a : b returns a if a is greater than b, else b.

---