All C Language Operators - English

Arithmetic Operators

Operator	Description	Example
+	Adds two operands	A + B = 30
-	Subtracts second operand from the first	A - B = -10
*	Multiplies both operands	A * B = 200
/	Divides numerator by de-numerator	B / A = 2
%	Modulus operator and remainder after integer division	B % A = 0
++	Increment operator increases the integer value by one	A++ = 11
--	Decrement operator decreases the integer value by one	A-- = 9

Assignment Operators

Operator	Description	Example
=	Simple assignment operator	C = A + B
+=	Add AND assignment operator	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator	C -= A is equivalent to C = C - A
*=	Multiply AND assignment operator	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator	C /= A is equivalent to C = C / A
%=	Modulus AND assignment operator	C %= A is equivalent to C = C % A
<<=	Left shift AND assignment operator	C <<= 2 is same as C = C << 2
>>=	Right shift AND assignment operator	C >>= 2 is same as C = C >> 2
&=	Bitwise AND assignment operator	C &= 2 is same as C = C & 2
^=	Bitwise exclusive OR and assignment operator	C ^= 2 is same as C = C ^ 2
|=	Bitwise inclusive OR and assignment operator	C |= 2 is same as C = C | 2

Relational (Comparison) Operators

Operator	Description	Example
==	Checks if values are equal	(A == B) is false
!=	Checks if values are not equal	(A != B) is true
>	Checks if left operand is greater than right	(A > B) is false
<	Checks if left operand is less than right	(A < B) is true
>=	Checks if left operand is greater than or equal to right	(A >= B) is false
<=	Checks if left operand is less than or equal to right	(A <= B) is true

Logical Operators

Operator	Description	Example
&&	Logical AND operator	(A && B) is false
||	Logical OR operator	(A || B) is true
!	Logical NOT operator (negation)	!(A && B) is true

Bitwise Operators

Operator	Description	Example
&	Binary AND operator	(A & B) will give 12, which is 0000 1100
|	Binary OR operator	(A | B) will give 61, which is 0011 1101
^	Binary XOR operator	(A ^ B) will give 49, which is 0011 0001
~	Binary ones complement operator	(~A) will give -61, which is 1100 0011
<<	Binary left shift operator	A << 2 will give 240, which is 1111 0000
>>	Binary right shift operator	A >> 2 will give 15, which is 0000 1111

Miscellaneous Operators

Operator	Description	Example
sizeof()	Returns the size of a variable in bytes	sizeof(a), where a is integer, will return 4
&	Returns the address of a variable	&a; returns the actual address of the variable
*	Pointer to a variable	*a;
? :	Conditional expression (ternary operator)	If Condition is true ? then value X : otherwise value Y
,	Comma operator for evaluating multiple expressions	a = (b = 3, b + 2); here a will be 5
.	Structure member access	struct_var.member
->	Structure pointer member access	struct_ptr->member
[]	Array subscripting	array[index]
()	Function call	function(parameters)
(type)	Type casting	(int)3.14

Operator Precedence (Priority Order)

1. Postfix: () [] -> . ++ --
2. Unary: + - ! ~ ++ -- (type)* & sizeof
3. Multiplicative: * / %
4. Additive: + -
5. Shift: << >>
6. Relational: < <= > >=
7. Equality: == !=
8. Bitwise AND: &
9. Bitwise XOR: ^
10. Bitwise OR: |
11. Logical AND: &&
12. Logical OR: ||
13. Conditional: ?:
14. Assignment: = += -= *= /= %= >>= <<= &= ^= |=
15. Comma: ,