Operator Precedence

Operator	Description	Sample Usage	Result	Associativity	Controls evaluation
( )	Grouping	(exp)	Same as exp	N/A	
( )	Function call	rexp (rexp, …, rexp)	rexp	L-R	No
[ ]	Subscript	rexp[rexp]	lexp	L-R	No
.	Structure member	lexp.member_name	lexp	L-R	No
->	Struc pointer member	rexp->member_name	lexp	L-R	No
++	Postfix increment	lexp++	rexp	L-R	No
--	Postfix decrement	lexp--	rexp	L-R	No
!	Logical negation	!rexp	rexp	R-L	No
~	One's complement	~rexp	rexp	R-L	No
+	Unary plus	+rexp	rexp	R-L	No
-	Unary minus	-rexp	rexp	R-L	No
++	Prefix increment	++lexp	rexp	R-L	No
--	Prefix decrement	--lexp	rexp	R-L	No
*	Indirection	*rexp	rexp	R-L	No
&	Address of	&lexp	rexp	R-L	No
sizeof	Size in bytes	sizeof rexp , sizeof (type)	rexp	R-L	No
(type)	Type conversion	type(rexp)	rexp	R-L	No
*	Multiplication	rexp * rexp	rexp	L-R	No
/	Division	rexp / rexp	rexp	L-R	No
%	Integer remainder	rexp % rexp	rexp	L-R	No
+	Addition	rexp + rexp	rexp	L-R	No
-	Subtraction	rexp - rexp	rexp	L-R	No
<<	Left shift	rexp << rexp	rexp	L-R	No
>>	Right shift	rexp >> rexp	rexp	L-R	No
>	Greater than	rexp > rexp	rexp	L-R	No
>=	Greater than or equal	rexp >= rexp	rexp	L-R	No
<	Less than	rexp < rexp	rexp	L-R	No
<=	Less than or equal	rexp <= rexp	rexp	L-R	No
==	Equal to	rexp == rexp	rexp	L-R	No
!=	Not equal to	rexp != rexp	rexp	L-R	No
&	Bitwise AND	rexp & rexp	rexp	L-R	No
^	Bitwise exclusive OR	rexp ^ rexp	rexp	L-R	No
|	Bitwise inclusive OR	rexp | rexp	rexp	L-R	No
&&	Logical AND	rexp / rexp	rexp	L-R	Yes
||	Logical OR	rexp / rexp	rexp	L-R	Yes
?:	Conditional	rexp ? rexp : rexp	rexp	N/A	Yes
=	Assignment	lexp = rexp	rexp	R-L	No
+=	Add to	lexp += rexp	rexp	R-L	No
-=	Subtract from	lexp -= rexp	rexp	R-L	No
*=	Multiply by	lexp *= rexp	rexp	R-L	No
/+	Divide by	lexp /= rexp	rexp	R-L	No
%=	Modulo by	lexp %= rexp	rexp	R-L	No
<<=	Shift left by	lexp <<= rexp	rexp	R-L	No
>>=	Shift right by	lexp >>= rexp	rexp	R-L	No
&=	AND with	lexp &= rexp	rexp	R-L	No
^=	Exclusive OR with	lexp ^= rexp	rexp	R-L	No
|=	Inclusive OR with	lexp |= rexp	rexp	R-L	No
,	Comma	rexp, rexp	rexp	L-R	Yes