

Chapter 1: Precedence of Operators

In the following table of operator precedence, the Turbo C++ operators are divided into 16 categories. **The #1 category has the highest precedence**; category #2 (Unary operators) takes second precedence, and so on to the Comma operator, which has lowest precedence. The operators within each category have equal precedence. **The Unary (category #2), Conditional (category #14), and Assignment (category #15) operators associate right-to-left; all other operators associate left-to-right.**

#	Category	Operator	What it is (or does)
1.	Highest	()	Function call
		[]	Array subscript
		->	C++ indirect component selector
		::	C++ scope access/resolution
		.	C++ direct component selector
2.	Unary	!	Logical negation (NOT)
		~	Bitwise (1's) complement
		+	Unary plus
		-	Unary minus
		++	Preincrement or postincrement
		--	Predecrement or postdecrement
		&	Address
		*	Indirection
		sizeof	(returns size of operand, in bytes)
		(type)	Type cast
		new	(dynamically allocates C++ storage)
		delete	(dynamically deallocates C++ storage)
3.	Multiplicative	*	Multiply
		/	Divide
		%	Remainder (modulus)
4.	Member access	.*	C++ dereference
		->*	C++ dereference
5.	Additive	+	Binary plus
		-	Binary minus
6.	Shift	<<	Shift left

	>>	Shift right
7. Relational	<	Less than
	<=	Less than or equal to
	>	Greater than
	>=	Greater than or equal to
8. Equality	==	Equal to
	!=	Not equal to
9.	&	Bitwise AND
10.	^	Bitwise XOR
11.		Bitwise OR
12.	&&	Logical AND
13.		Logical OR
14. Conditional	?:	(a ? x : y means "if a then x, else y")
15. Assignment	=	Simple assignment
	*=	Assign product
	/=	Assign quotient
	%=	Assign remainder (modulus)
	+=	Assign sum
	-=	Assign difference
	&=	Assign bitwise AND
	^=	Assign bitwise XOR
	=	Assign bitwise OR
	<<=	Assign left shift
	>>=	Assign right shift
16. Comma	,	Evaluate

All of the operators in this table can be overloaded except the following:

- . C++ direct component selector
- * C++ dereference
- :: C++ scope access/resolution
- ?: Conditional