1	::	Scope resolution operator	left to right
2	()	Function call	left to right
	()	Member initalization	
	[]	Array access	
	->	Member access from a pointer	
	.	Member access from an object	
	++	Post-increment	
	--	Post-decrement	
	dynamic_cast	Runtime-checked type conversion	
	static_cast	Unchecked type conversion	
	reinterpret_cast	Reinterpreting type conversion	
	const_cast	Cast away/Add constness	
	typeid	Get type information	
3	!	Logical negation	right to left
	not	Alternate spelling for !	
	~	Bitwise complement	
	compl	Alternate spelling for ~	
	++	Pre-increment	
	--	Pre-decrement	
	-	Unary minus	
	+	Unary plus	
	*	Dereference	
	&	Address of	
	sizeof	Size (of the type) of the operand in bytes	
	new	Dynamic memory allocation	
	new []	Dynamic memory allocation of array	
	delete	Deallocating the memory	
	delete []	Deallocating the memory of array	
	(type)	Cast to a given type	
4	->*	Member pointer selector	left to right
	.*	Member object selector	
5	*	Multiplication	left to right
	/	Division	
	%	Modulus	
6	+	Addition	left to right
	-	Subtraction	
7	<<	Bitwise shift left	left to right
	>>	Bitwise shift right	
8	<	Comparison less-than	left to right
	<=	Comparison less-than-or-equal-to	
	>	Comparison greater-than	
	>=	Comparison greater-than-or-equal-to	
9	==	Comparison equal-to	left to right
	eq	Alternate spelling for ==	
	!=	Comparison not-equal-to	
	not_eq	Alternate spelling for !=	
10	&	Bitwise AND	left to right
	bitand	Alternate spelling for &	
11	^	Bitwise exclusive OR (XOR)	left to right
	xor	Alternate spelling for ^	
12	|	Bitwise inclusive (normal) OR	left to right
	bitor	Alternate spelling for |	
13	&&	Logical AND	left to right
	and	Alternate spelling for &&	
14	||	Logical OR	left to right
	or	Alternate spelling for ||	
15	? :	Ternary conditional (if-then-else)	right to left
16	=	Assignment operator	right to left
	+=	Increment and assign	
	-=	Decrement and assign	
	*=	Multiply and assign	
	/=	Divide and assign	
	%=	Modulo and assign	
	&=	Bitwise AND and assign	
	and_eq	Alternate spelling for &=	
	^=	Bitwise exclusive or (XOR) and assign	
	xor_eq	Alternate spelling for ^=	
	|=	Bitwise normal OR and assign	
	or_eq	Alternate spelling for |=	
	<<=	Bitwise shift left and assign	
	>>=	Bitwise shift right and assign	
17	throw	throw exception	
18	,	Sequential evaluation operator	left to right