

## Operators 重写规则

There are a few more interesting operators.

Arithmetic	+	-	*	/	%		
	+=	-=	*=	/=	%=		
Bitwise		&		~	!		
Relational	==	!=	<	>	<=	>=	<=>
Stream	<<	>>	<<=	>>=			
Logical	&&		^	&=	=	^=	
Increment	++	--					
Memory	->	->*	new	new []	delete	delete []	
Misc	( )	[ ]	,	=		co_await	

1. Some operators must be implemented as members (eg. [], (), ->, =) due to C++ semantics.
2. Some must be implemented as non-members (eg. <<, >>, if you are writing class for rhs, not lhs).
3. If unary operator (eg. ++), implement as member.
4. If binary operator and treats both operands equally (eg. both unchanged) implement as non-member (maybe friend). Examples: +, <.
5. If binary operator and not both equally (changes lhs), implement as member (allows easy access to lhs private members). Examples: +=