

# **Objektorientiertes Programmieren mit C++ für Fortgeschrittene**

## **ANHANG**

### **Präcedenztabelle der Operatoren**

### Präzedenztablelle der Operatoren

Die folgende Tabelle gibt Auskunft über den Vorrang von Operatoren gegenüber anderen. Je niedriger die Rangnummer desto höher der Vorrang in der Auswertung. Operatoren mit gleicher Rangnummer besitzen gleiche Präzedenz. Die **Assoziativität** gibt an in welcher Reihenfolge die Operatoren gleicher Präzedenz ausgewertet werden.

Rang	Operatoren		Assoziativität
1	:: ( )	Scope Resolution Operator (binär und unär) Klammerung	links → rechts
2	. -> [] functionname() Lval++ Lval-- typeid()  type() dynamic_cast<type>() static_cast<type>() reinterpret_cast<type>() const_cast<type>()	Objektelement-Operatoren Indexoperator Funktionsaufruf Postfix-Operatoren Typobjekt  Wertkonstruktion/ Typkonvertierungen	links → rechts
3	new new[] delete delete[] sizeof ++Lval --Lval ~ ! - + & * (type)	Dynamische Speicherallokation Dynamische Speicherdeallokation Objekt-/Typgrößenermittlung Präfix-Operatoren Unäre Modifikatoren Adressermittlung/Dereferenzierung C-Typkonvertierung	rechts → links
4	->* .*	Dereferenzierte Pointer auf Klassenkomponenten (Komponentenauswahloperatoren)	links → rechts
5	* / %	Arithmetische Operatoren	links → rechts
6	+ -	Arithmetische Operatoren	links → rechts
7	<< >>	Schiebeoperatoren	links → rechts
8	< <= > >=	Vergleichsoperatoren	links → rechts
9	== !=	Vergleichsoperatoren	links → rechts
10	&	Binäres Und	links → rechts
11	^	Binäres Exklusiv-Oder	links → rechts
12		Binäres Oder	links → rechts
13	&&	Logisches Und	links → rechts
14		Logisches Oder	links → rechts
15	? : (1. Operand)	bedingter Auswerteoperator	links → rechts
16	= *= /= %= += -= <<= >>= &=  = ^=	Zuweisungsoperatoren	rechts → links
17	throw	Werfen von Ausnahmen	links → rechts
18	,	Sequentieller Auswerteoperator	links → rechts
19	? : (2. und 3. Operand)	bedingter Auswerteoperator	links → rechts

- ◇ Die Reihenfolge der Auswertung von **Operanden eines Operators** ist undefiniert!
- ◇ Die Tabelle ist nicht Teil des ISO-C++-Standards. Sie ergibt sich aus den festgelegten Grammatikregeln.
- ◇ Das logische Und (&&) und das logische Oder (||) werden nur so weit erforderlich von links nach rechts ausgewertet.