

## Appendice – Priorità degli operatori (1/2)

<b>Operatore (in grassetto)</b>	<b>Nome</b>	<b>Ass.</b>
<b>class-name :: member</b> <b>namespace-name :: member</b>	risolutore di visibilità	s
<b>:: member</b>	risolutore globale di visibilità	d
<b>object . member</b>	selezione	s
<b>pointer -&gt; member</b>	dereferenziazione e selezione	s
<b>array [ rvalue ]</b>	indicizzazione	s
<b>function ( actual-argument-list )</b>	chiamata di funzione	s
<b>lvalue ++</b>	postincremento	d
<b>lvalue --</b>	postdecremento	d
<b>static_cast type &lt;rvalue&gt;</b>	conversione di tipo	s
<b>const_cast type &lt;rvalue&gt;</b>	conversione const	s
<b>sizeof object / sizeof (type)</b>	dimensione di oggetto o di tipo	d
<b>++ lvalue</b>	preincremento	d
<b>-- lvalue</b>	predecremento	d
<b>~ rvalue</b>	complemento bit a bit	d
<b>! rvalue</b>	negazione	d
<b>- rvalue</b>	meno unario	d
<b>+ rvalue</b>	più unario	d
<b>&amp; lvalue</b>	indirizzo	d
<b>* rvalue</b>	dereferenziazione	d
<b>new type</b>	allocazione	d
<b>delete pointer</b>	deallocazione	d
<b>delete[] pointer</b>	deallocazione di array	d

## Appendice – Priorità degli operatori (2/2)

Operatore	Nome	Ass.
object .* pointer-to-member	selezione con punt. a membro	s
pointer ->* pointer-to-member	deref. e sel. con punt. a membro	s
rvalue * rvalue	moltiplicazione	s
rvalue / rvalue	quoziente	s
rvalue % rvalue	resto	s
rvalue + rvalue	somma	s
rvalue - rvalue	sottrazione	s
rvalue << rvalue	traslazione sinistra	s
rvalue >> rvalue	traslazione destra	s
rvalue < rvalue	minore	s
rvalue <= rvalue	minore o uguale	s
rvalue > rvalue	maggiore	s
rvalue >= rvalue	maggiore o uguale	s
rvalue == rvalue	uguale	s
rvalue != rvalue	diverso	s
rvalue & rvalue	AND bit a bit	s
rvalue ^ rvalue	OR esclusivo bit a bit	s
rvalue   rvalue	OR bit a bit	s
rvalue && rvalue	AND logico	s
rvalue    rvalue	OR logico	s
rvalue ? rvalue : rvalue	espressione condizionale	s
lvalue = rvalue	assegnamento	d
lvalue += rvalue	somma e assegnamento	d
lvalue -= rvalue	sottrazione e assegnamento	d
lvalue *= rvalue	moltiplicazione e assegnamento	d
lvalue /= rvalue	divisione e assegnamento	d
lvalue %= rvalue	resto e assegnamento	d
lvalue &= rvalue	AND bit a bit e assegnamento	d
lvalue  = rvalue	OR bit a bit e assegnamento	d
lvalue ^= rvalue	OR esclusivo bit a bit e assegnamento	d
lvalue <<= rvalue	traslazione a sinistra e assegnamento	d
lvalue >>= rvalue	traslazione a destra e assegnamento	d
expression , expression	virgola	s