

Operatoren

<code>...->...</code>	Auf einen Member eines Objekts zugreifen , auf das ein Pointer gegeben ist.
<code>ptr->mem</code> macht das Selbe wie <code>(*ptr).mem</code> .	
<pre>struct my_class { // POST: "Hi! " is written to std::cout void say_hi () const { std::cout << "Hi! "; } }; ... my_class obj; my_class* ptr = &obj; // just a pointer to obj obj.say_hi(); // direct access ptr->say_hi(); // using -> (*ptr).say_hi(); // equivalent</pre>	

<code>... ? ... : ...</code>	Kurzes if/else mit "Rückgabewert"
<code>... ? ... : ...</code> macht den Code oft deutlich kompakter, als wenn das Selbe mit einem <code>if/else</code> erzielt wird. Es ist beispielsweise nützlich, wenn einer Variable basierend auf einer Condition unterschiedliche Werte zugewiesen werden sollen. Hingegen kann sowohl bei <code>...</code> als auch bei <code>...</code> nur jeweils ein Ausdruck ausgewertet werden (i.e. keine Blöcke <code>{...}</code> , keine Variablendefinitionen, ...).	
<pre>int z; std::cin >> z; const int z_abs = (z < 0) ? -z : z; // User enters 3 --> z_abs is 3 // User enters -3 --> z_abs is 3</pre>	

Zur **Vererbung** wird kein Summary publiziert.