

## Grundgerüst

Grundgerüst	Grundgerüst für einfache Programme
<p>Unser Grundgerüst besteht aus folgenden Komponenten:</p> <ul style="list-style-type: none"><li>• <b>Kommentare</b></li><li>• <b>Verfügbarmachung zusätzlicher Funktionalität</b> (via <code>#include &lt;...&gt;</code>)</li><li>• <b>main-Funktion</b></li></ul> <p>Beachte: Jedes ausführbare Programm braucht die <b>main-Funktion</b>.</p> <p>Beachte: <b>Zusätzliche Funktionalität</b>, wie beispielsweise Eingabe und Ausgabe, ist nicht in der "Grundfunktionalität" von C++ enthalten. Der Programmierer muss also mitteilen, wo diese zusätzliche "Funktionalität" definiert ist. Mittels <code>#include &lt;...&gt;</code> kann er dem Compiler sagen, in welcher Befehlssammlung (genannt: Library) diese "Funktionalität" definiert ist. Ein- und Ausgabe sind beispielsweise in <code>iostream</code> definiert.</p>	
<pre>// Informatik - Serie 13 - Aufgabe 4e // Programm: my_program.cpp // Autor: X. M. Mueller (Gruppe F)  #include &lt;iostream&gt;  int main () {     // Your code here...      return 0; }</pre>	

## Datentypen

<code>int</code>	Datentyp für <b>ganze Zahlen</b>
<pre>int a = 3; int b = a + 4;</pre>	

## Operatoren

# Programmier-Befehle - Woche 1

<b>*</b>	<b>Multiplikation</b> von zwei <i>R</i> -Werten.
Präzedenz: 14 und Assoziativität: <b>links</b>	
<pre>int a = -3 * 4; int b = 2 * a; // Note: use l-value as r-value</pre>	

<b>=</b>	<b>Zuweisungsoperator.</b> Weist einem L-Wert einen neuen Wert zu.
Präzedenz: 4 und Assoziativität: <b>rechts</b>	
<pre>int a; a = 4; // value 4 a = 3; // value 3</pre>	

<b>&gt;&gt;</b>	(siehe: <code>std::cin</code> unter "Input/Output")
Präzedenz: 12 und Assoziativität: <b>links</b>	

<b>&lt;&lt;</b>	(siehe: <code>std::cout</code> unter "Input/Output")
Präzedenz: 12 und Assoziativität: <b>links</b>	

## Input/Output

<code>std::cin &gt;&gt; ...</code>	<b>Eingabe</b> via Terminal (z.B. Tastatureingabe)
Erfordert: <code>#include&lt;iostream&gt;</code>	
<pre>int a; std::cin &gt;&gt; a; // stores the user input into a</pre>	

<code>std::cout &lt;&lt; ...</code>	<b>Ausgabe</b> in das Terminal
Erfordert: <code>#include&lt;iostream&gt;</code>	
Das <code>\n</code> bewirkt einen Zeilenumbruch. Man kann stattdessen auch <code>std::endl</code> verwenden (siehe Beispiel unten).	
<pre>// tell the user to enter a number std::cout &lt;&lt; "Enter height in metres: "; int h; std::cin &gt;&gt; h; std::cout &lt;&lt; "Your input was: " &lt;&lt; h &lt;&lt; " m\n";  // line breaks std::cout &lt;&lt; "This text is output..." &lt;&lt; std::endl; std::cout &lt;&lt; "This text is output...\n"; // does the same std::cout &lt;&lt; "Twice the same was output...\n";</pre>	