

Grundgerüst

Grundgerüst	Grundgerüst für einfache Programme
<p>Unser Grundgerüst besteht aus folgenden Komponenten:</p> <ul style="list-style-type: none">• Kommentare• Verfügbarmachung zusätzlicher Funktionalität (via <code>#include <...></code>)• main-Funktion <p>Beachte: Jedes ausführbare Programm braucht die main-Funktion.</p> <p>Beachte: Zusätzliche Funktionalität, 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 <...></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 <iostream> int main () { // Your code here... std::cout << "Hello World!\n"; return 0; }</pre>	

Datentypen

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

Programmier-Befehle - Woche 01

<code>const ...</code>	Schreibzugriff auf Variable verboten
Gemeint ist natürlich der Schreibzugriff <i>nach</i> der Initialisierung.	
<pre>int a = 3; const int b = 4; a = 5; // valid b = 3; // not valid since b is const int c = -2 * b; // valid since just WRITE-access is forbidden // by "const"</pre>	

Operatoren

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

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

<code>>></code>	(siehe: <code>std::cin</code> unter "Input/Output") (...)
-----------------------	--

(...)

Präzedenz: 12 und Assoziativität: links

<<

(siehe: `std::cout` unter "Input/Output")

Präzedenz: 12 und Assoziativität: links

Input/Output

`std::cin >> ...`

Eingabe via Terminal (z.B. Tastatureingabe)

Erfordert: `#include <iostream>`

```
int a;
std::cin >> a; // stores the user input into a
```

`std::cout << ...`

Ausgabe in das Terminal

Erfordert: `#include <iostream>`

Das `\n` bewirkt einen Zeilenumbruch.

```
// tell the user to enter a number
std::cout << "Enter height in metres: ";
int h;
std::cin >> h;
std::cout << "Your input was: " << h << " m\n";
```