

Übung01: Informationstechnik (IT)

Institutsleitung
Prof. Dr.-Ing. K. D. Müller-Glaser
Prof. Dr.-Ing. J. Becker
Prof. Dr. rer. nat. W. Stork

Tobias Schwalb & Timo Sandmann & Stephan Werner

Institut für Technik der Informationsverarbeitung (ITIV)



Hello World, Variablen und Operatoren

KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

www.kit.edu

Lehrpersonen

■ Vorlesung

- Prof. Dr.-Ing. Klaus D. Müller-Glaser
 - Gebäude 30.10 - Raum 336
 - Telefon: 608 - 42500
 - klaus.mueller-glaser@kit.edu
 - Sprechstunde: Mittwoch 11:30 bis 12:30
und nach Vereinbarung



■ Übungen und Tutorium

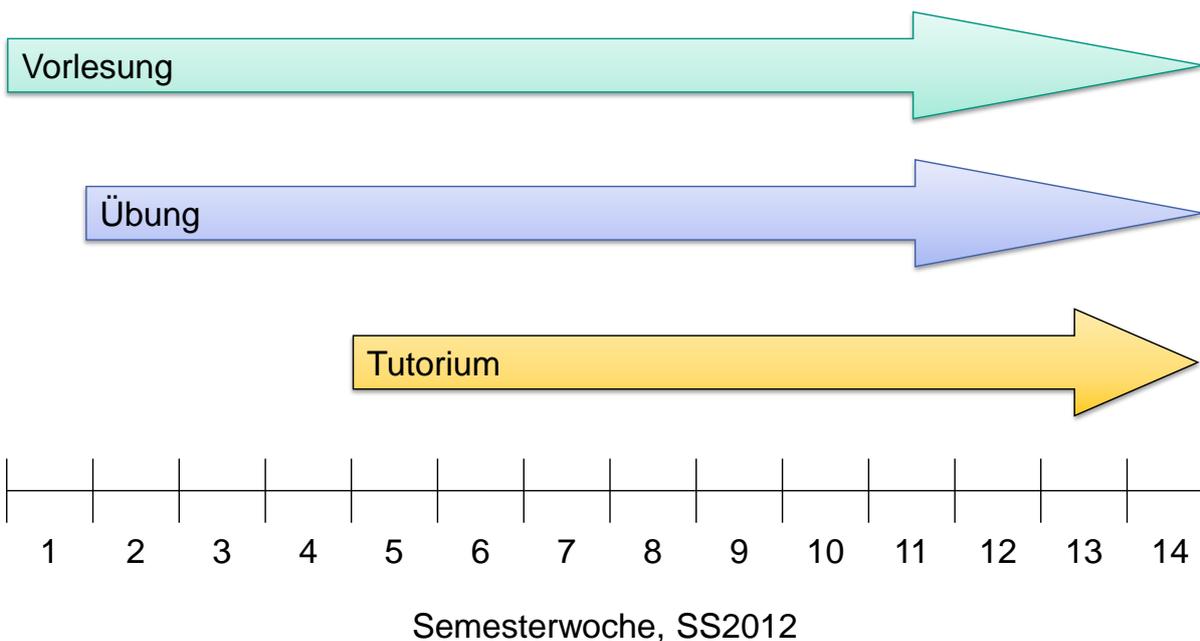
- Dipl.-Ing. (FH) Tobias Schwalb
 - tobias.schwalb@kit.edu
 - Geb. 30.10 - Raum 227, Tel. 608 - 43910
- Dipl.-Inf. Timo Sandmann
 - timo.sandmann@kit.edu
 - Geb. 30.10 - Raum 226.2, Tel. 608 - 41939
- Dipl.-Ing. Stephan Werner
 - stephan.werner@kit.edu
 - Geb. 30.10 - Raum 226.2, Tel. 608 - 47169
- Sprechstunden nach Vorlesung/Übung und nach Vereinbarung



Inhalt: Übung01

- 1 • Organisatorisches
- 2 • Einführung
- 3 • Erstellen eines C++ Programms
- 4 • Variablen
- 5 • Operatoren

Aufteilung Veranstaltung



IT-Veranstaltungsplan

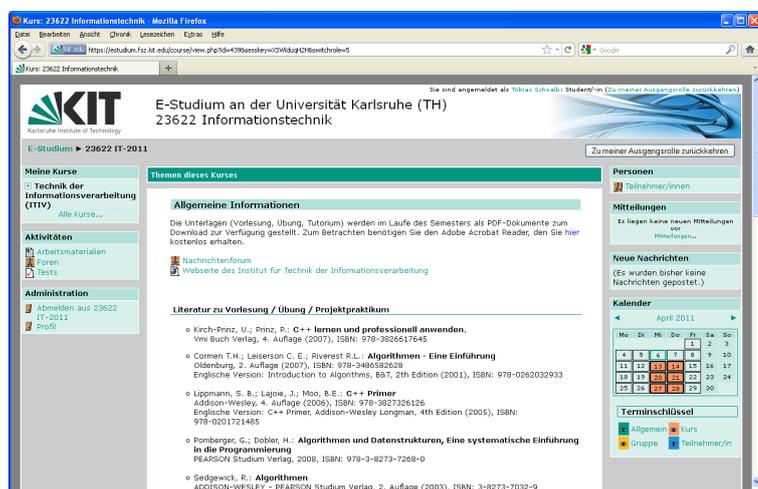
SW	Datum, Mi	09:45 – 11:15	Datum, Do	14:00 – 15:30	Tutorium
1.	18. Apr.	Vorlesung	19. Apr.	Vorlesung	
2.	25. Apr.	Vorlesung	26. Apr.	Übung	
3.	02. Mai	Vorlesung	03. Mai	Übung	
4.	09. Mai	Vorlesung	10. Mai	Vorlesung	
5.	16. Mai	Übung	17. Mai	Feiertag	Tutorium
6.	23. Mai	Vorlesung	24. Mai	Übung	Tutorium
7.	30. Mai	Pfingsten	31. Mai	Pfingsten	Tutorium
8.	06. Jun.	Übung	07. Jun.	Feiertag	Tutorium
9.	13. Jun.	Vorlesung	14. Jun.	Vorlesung	Tutorium
10.	20. Jun.	Vorlesung	21. Jun.	Übung	Tutorium
11.	27. Jun.	Vorlesung	28. Jun.	Vorlesung	Tutorium
12.	04. Jul.	Übung	05. Jul.	Vorlesung	Tutorium
13.	11. Jul.	Vorlesung	12. Jul.	Übung	Tutorium
14.	18. Jul.	Probeklausur	19. Jul.	Vorlesung	Tutorium

SW = Semesterwoche

Unterlagen zu Informationstechnik

- Nutzung der Lernplattform eStudium (Moodle)
 - Link: <https://estudium.fsz.kit.edu/>
 - Registrierung notwendig (falls noch nicht erfolgt)
 - Einschreibung zu: 23622 Informationstechnik

- Inhalt
 - Vorlesungsfolien
 - Übungsblätter
 - Übungsfolien
 - Tutoriumsaufgaben
 - Foren
 - Aufgabenabgabe
 - ...

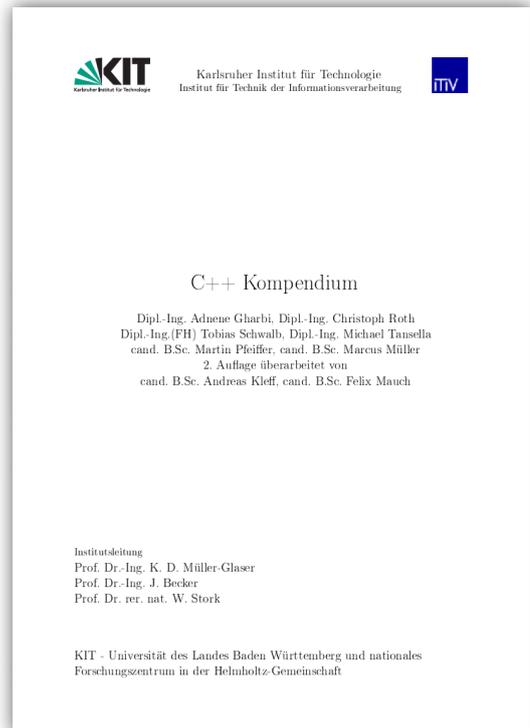


The screenshot shows the Moodle interface for course 23622 Informationstechnik. The main content area is titled 'Literatur zu Vorlesung / Übung / Projektpraktikum' and lists several books and papers:

- Kirch-Prinz, U.; Prinz, P.: **C++ lernen und professionell anwenden**, Vms Buch Verlag, 4. Auflage (2007), ISBN: 978-3825617645
- Cormen T.H.; Leiserson C. E.; Rivest R.L.; **Algorithmen - Eine Einführung** (Übersetzung, 2. Auflage (2007), ISBN: 978-3486524508
Englische Version: Introduction to Algorithms, 8&T, 2th Edition (2001), ISBN: 978-0262032993
- Lippmann, S. B.; Lajoie, J.; Moo, B.E.: **C++ Primer** Addison-Wesley, 4. Auflage (2006), ISBN: 978-027326126
Englische Version: C++ Primer, Addison-Wesley Longman, 4th Edition (2001), ISBN: 978-0201721485
- Pomberger, G.; Doblar, H.: **Algorithmen und Datenstrukturen, Eine systematische Einführung in die Programmierung** Pearson Studium Verlag, 2008, ISBN: 978-3-8273-7269-0
- Sedgewick, R.: **Algorithmen** Addison-Wesley - Pearson Studium Verlag, 2. Auflage (2003), ISBN: 3-8273-7032-9

C++ Kompendium

- Skript zur Einführung in die Programmiersprache C++
- Begleitend zur Übung
- Fokus auf eine einfache und verständliche Darstellung
- Behandelt nicht alle Themen und Möglichkeiten von C++ vollständig
- Enthält Verweise auf weiterführende Literatur (u.a. Aufgaben)
- In Zusammenarbeit mit Studenten entwickelt



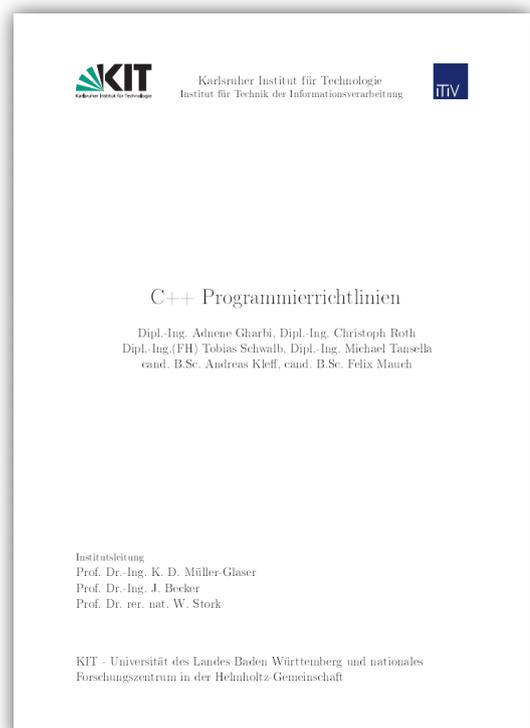
Ü1-7 26.04.2012

T. Schwalb, T. Sandmann, S. Werner - Informationstechnik (IT) - Hello World, Variablen und Operatoren

© Institut für Technik der Informationsverarbeitung (ITIV)

C++ Programmierrichtlinien

- Richtlinien zur Programmierung in C++
- Hilft:
 - Mehr Übersichtlichkeit
 - Fehler vermeiden
 - Einfachere Arbeit eines Teams an gleichem Projekt
 - Bessere Wartbarkeit des Codes
- Aufbau entsprechend dem Kompendium
- In Zusammenarbeit mit Studenten entwickelt



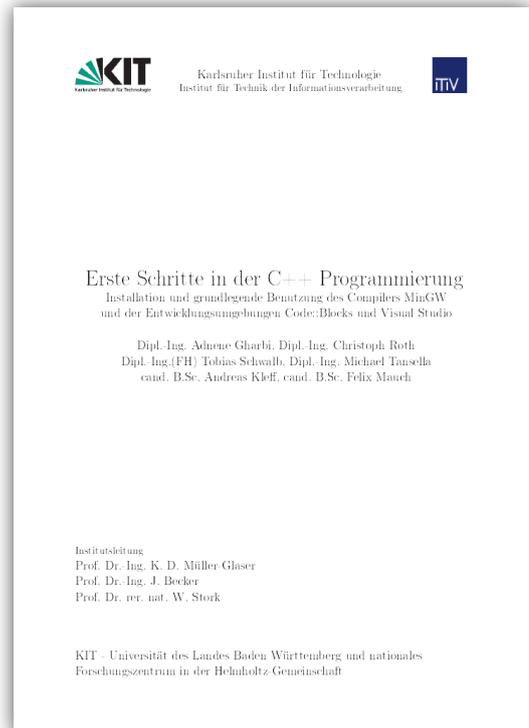
Ü1-8 26.04.2012

T. Schwalb, T. Sandmann, S. Werner - Informationstechnik (IT) - Hello World, Variablen und Operatoren

© Institut für Technik der Informationsverarbeitung (ITIV)

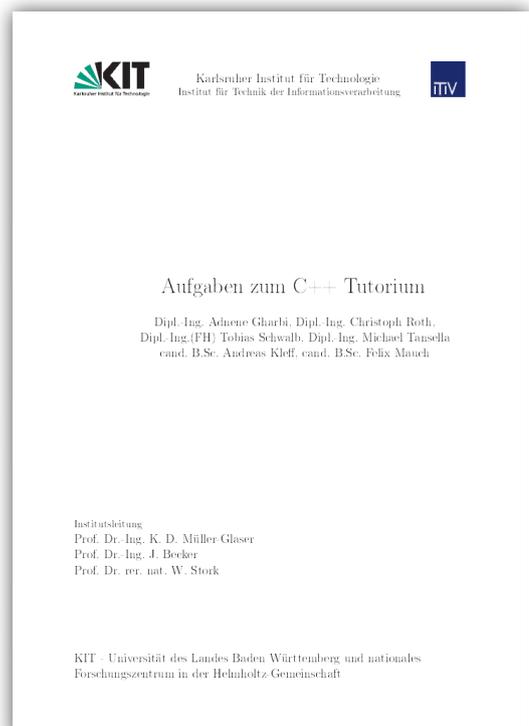
Erste Schritte in der C++ Programmierung

- Installation und grundlegende Benutzung von Compilern und Entwicklungsumgebungen
- Behandelt:
 - MinGW
 - CodeBlocks
 - Visual Studio
- Fokus auf eine einfache und verständliche Darstellung
- In Zusammenarbeit mit Studenten entwickelt

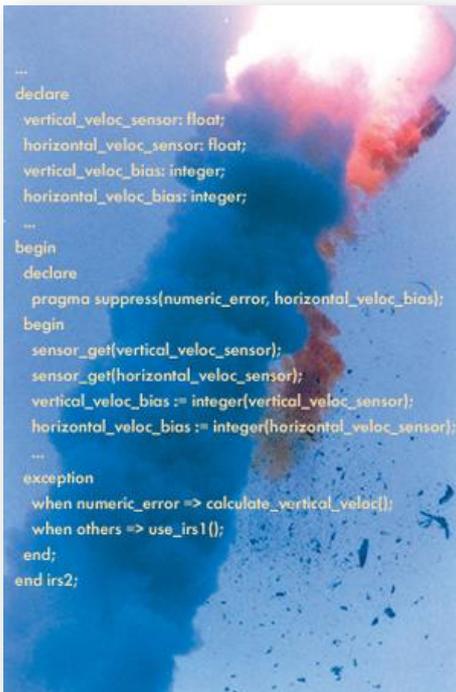


Aufgaben zum C++ Tutorium

- Sammlung mit C++ Programmieraufgaben
- Bearbeitung im Tutorium oder selbstständig
- Enthält Verweise auf das Kompendium und die Richtlinien
 - Aufbau in Bezug auf das Kompendium (steigende Komplexität)
- In Zusammenarbeit mit Studenten entwickelt



Warum Informationstechnik?



Quelle: <http://www.aero.org/publications/crosslink/fall2005/06.html>

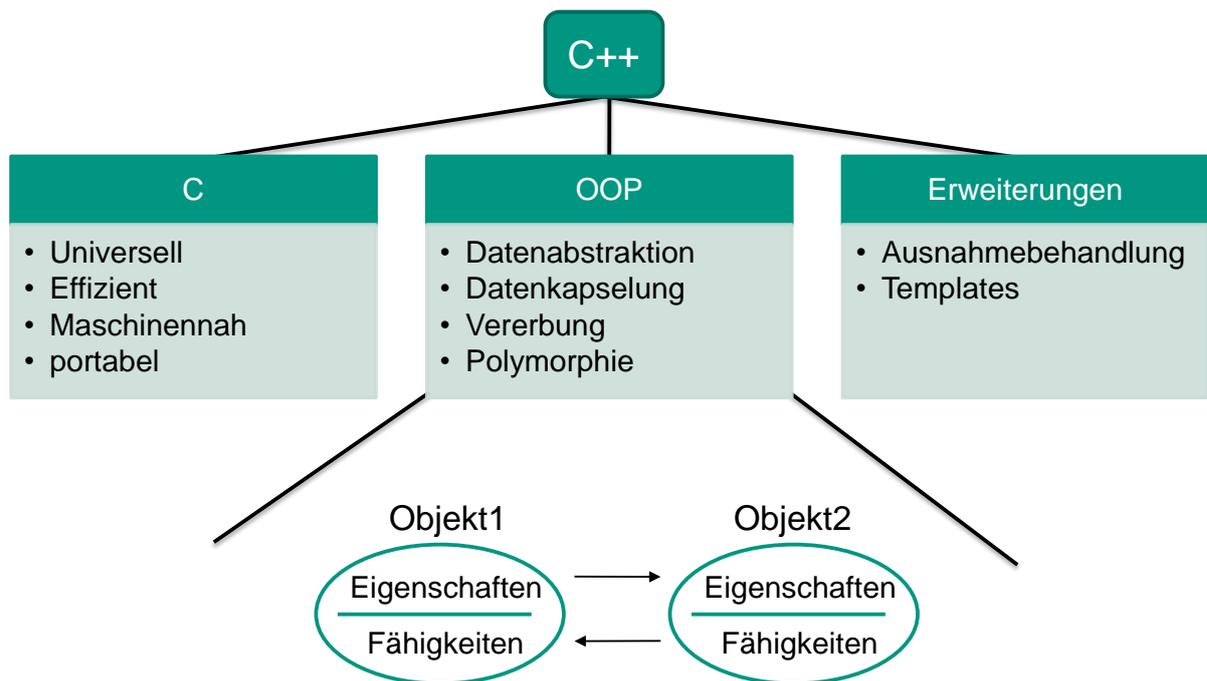
- Absturz der Ariane 5
 - Erstflug am 4. Juni 1996
- Grund:
 - Überlauf der Variablen im Lenksystem
 - Umwandlung einer 64-Bit-Gleitkommazahl in eine vorzeichenbehaftete 16-Bit-Ganzzahl
 - Absturz des Lenksystems
- Schaden:
 - 500 Mio. Euro

Informationstechnik Übung

Exemplarische Einführung der höheren Programmiersprache C++

- C und C++ sind die derzeit am weitest verbreiteten Programmiersprachen bezogen auf die Zahl der Anwendungen
- C ist hardwarenah (Treiberprogrammierung)
- C++ umfangreiche Erweiterung von C
- C++ exemplarisch für objektorientierte Programmierung (kommt erst später in der Vorlesung)
- C++ ist eine der mächtigsten Programmiersprachen
- C++ auch Basis für die Hardwarebeschreibungssprache SystemC

Warum C++?



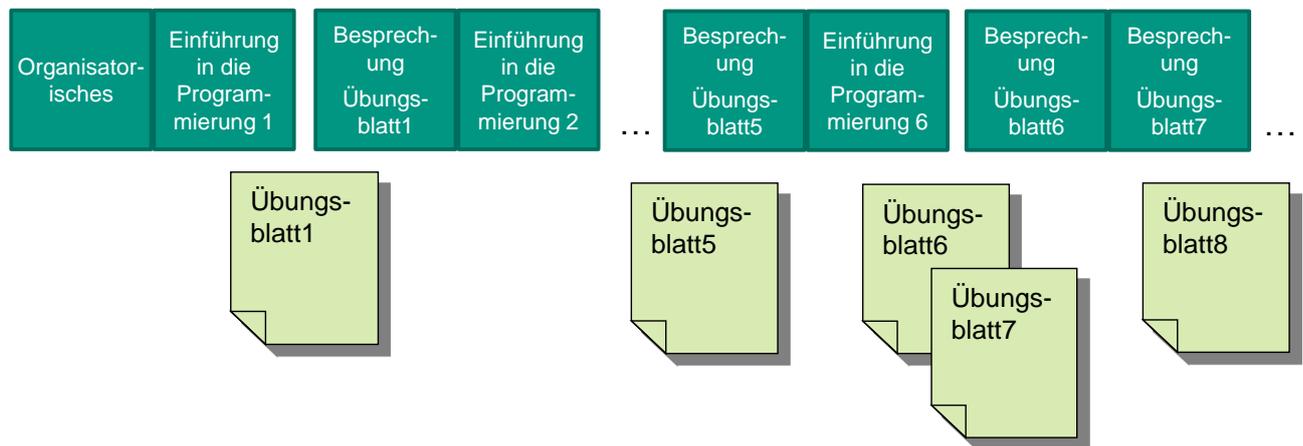
Lernziele der Übung

- Der Lernende soll am Ende:
 - Die Grundzüge der Programmiersprache C++ anwenden können
 - Konkrete Problemstellungen mit Hilfe der Informationstechnik lösen können
 - Seine Programme nach den Prinzipien der Objektorientierung strukturiert aufbauen können
 - Algorithmen in unterschiedlichen Darstellungsformen beschreiben können und in lauffähige Programme umsetzen können
 - Qualitätsmerkmale von Algorithmen und Programmen kennen und anwenden und hierzu Tests und Testprogramme erstellen
 - Mit den Grundfunktionen einer Entwicklungsumgebung umgehen können



Teilung der Übung

- Übung besteht im Allgemeinen aus 2 Teilen
 - 1. Teil: Besprechung der Übungsaufgaben
 - 2. Teil: Einführung in die Programmierung



Methoden

- Übungsaufgaben
- Powerpoint Präsentation
- Tafelarbeit / Overheadprojektor
- Live Vorführungen / Simulationen
- Methoden des aktiven Lernens
- Parallelität zum C++ Kompendium



Aktives Lernen

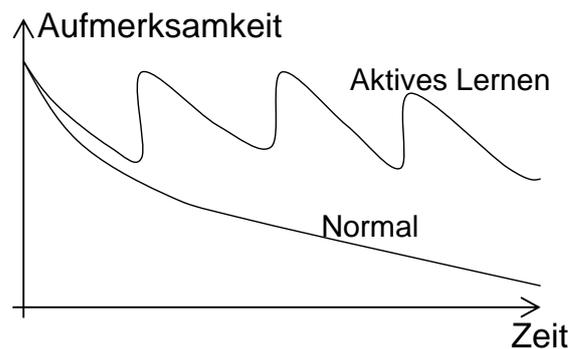
Aktive / Direkte Mitarbeit der Studenten

- Kurze Aufgaben während der Übung zum direkten Lösen
- Max. 2 Minuten
- In Gruppenarbeit
- Direkte Lösungsanfrage an Gruppen
- Themenwechsel



Ziele

- Schnelleres Lernen
- Weniger Nacharbeit
- Direkteres Verstehen
- Erhöhte Aufmerksamkeit

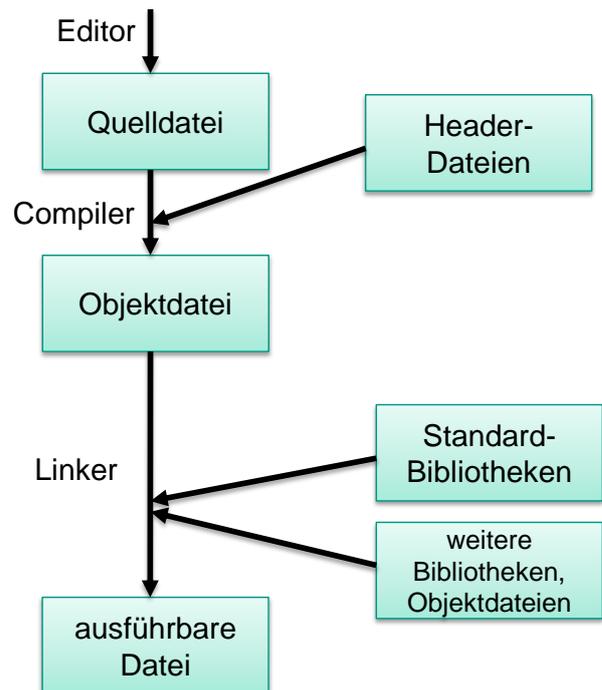


Inhaltsverzeichnis Übung

1	• Organisatorisches, Einführung, Aufbau • Erste Schritte in C++ & „Hello World“ & Variablen & Operatoren
2	• Arrays • Kontrollstrukturen
3	• Funktionen und Gültigkeitsbereiche • Zeiger
4	• Objektorientierung • Dynamische Speicherverwaltung
5	• Vererbung und Polymorphie • Erweiterte Datentypen & Strings
6	• Dateiverarbeitung • Verkettete Listen
7	• Such- und Sortieralgorithmen • Laufzeitanalyse
8	• Optimierungsalgorithmen • Rechnerarchitektur

Erstellen eines C++ Programms

- Editor dient zur Eingabe des Programmcodes
- Mehrere Quelldateien & Headerdateien sind möglich
- Objektdatei enthält den Maschinencode
- Linker führt alle Bausteine zur einer ausführbaren Datei zusammen
- Übliche Dateiendungen:
 - *.cpp >> Quelldatei
 - *.h >> Headerdatei
 - *.obj >> Objektdatei
 - *.exe >> ausführbare Datei



Erstes C++ Programm

```
// Mein erstes C++ Programm
```

Kommentar bis zum Zeilenende mit //

```
#include <iostream>
```

kopiert die Datei <iostream> an diese Stelle
→ C++ Header-Datei für Ein- und Ausgabe

```
using namespace std;
```

Definiert den vorgegebenen Namensbereich
Standard zu benutzen

```
int main()
```

Ein C++ Programm beginnt immer mit der ersten
Anweisung in der Funktion `main()`

```
{
```

Mehrzeilige Kommentare zwischen /* und */

```
/* Die ganze Funktionalität dieses Programms */
```

```
cout << "Hello World" << endl;
```

```
return 0;
```

Befehle werden mit Semikolon abgeschlossen

```
}
```

Ausgabe auf die Konsole

```
cout << "Hello World" << endl;
```

■ Ausgabefunktion

- `cout` = console output
- `<<` = Zeichen in den Ausgabestrom schieben
- `endl` = end of line
- Text steht in doppelten Anführungszeichen



Achtung: Ausgabe bestimmter Zeichen nur über Escape-Sequenzen (siehe Kompendium) möglich

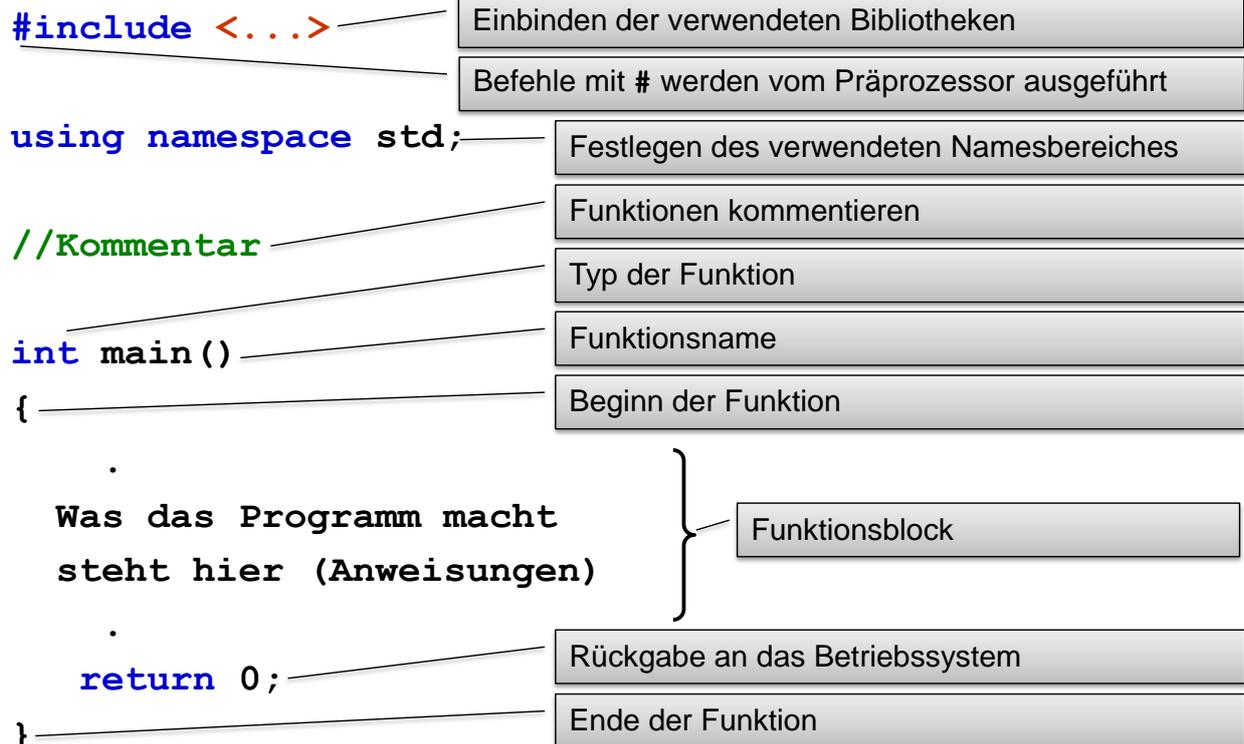
```

C:\> Eingabeaufforderung
D:\>cd code
D:\code>cd cpp
D:\code\cpp>cd hello
D:\code\cpp\hello>g++ -o HelloWorld.exe hello.cpp
D:\code\cpp\hello>HelloWorld.exe
Hello World
D:\code\cpp\hello>_
  
```

Escape-Sequenzen

Einzelzeichen	Bedeutung	ASCII-Wert (dezimal)
<code>\a</code>	alert (BEL)	7
<code>\b</code>	backspace (BS)	8
<code>\t</code>	horizontal tab (HT)	9
<code>\n</code>	line feed (LF)	10
<code>\v</code>	vertikal tab (VT)	11
<code>\f</code>	form feed (FF)	12
<code>\r</code>	carriage return (CR)	13
<code>\"</code>	"	34
<code>\'</code>	'	39
<code>\?</code>	?	63
<code>\\</code>	\	92
<code>\0</code>	Stringende-Zeichen	0
<code>\ooo</code> (bis zu drei Oktalziffern)	numerischer Wert eines Zeichens	ooo (oktal!)
<code>\xhh</code> (Folge von Hex-Ziffern)	numerischer Wert eines Zeichens	hh (hexadezimal!)

Aufbau eines Programms



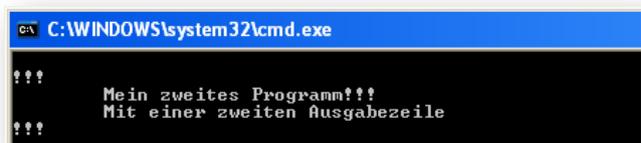
Zwischenübung01: Prog. Aufbau

Ordnen Sie die folgenden Zeilen in der richtigen Reihenfolge, sodass sich ein lauffähiges Programm mit folgender Ausgabe ergibt



```

01 return 0;
02 cout << endl;
03 int main()
04 cout << "!!!" << endl;
05 {
06 //Dies ist die Hauptfunktion
07 #include <iostream>
08 cout << endl << "!!!";
09 cout << "\tMein zweites Programm";
10 using namespace std;
11 }
12 /* Die ganze Funktionalität dieses Programms */
13 cout << "\tMit einer zweiten Ausgabezeile" << endl;
14 cout << "!!!" << endl;
  
```



```

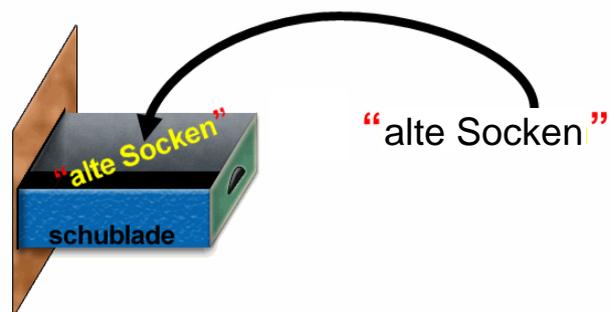
C:\WINDOWS\system32\cmd.exe
!!!
Mein zweites Programm!!!
Mit einer zweiten Ausgabezeile
  
```

Daten speichern

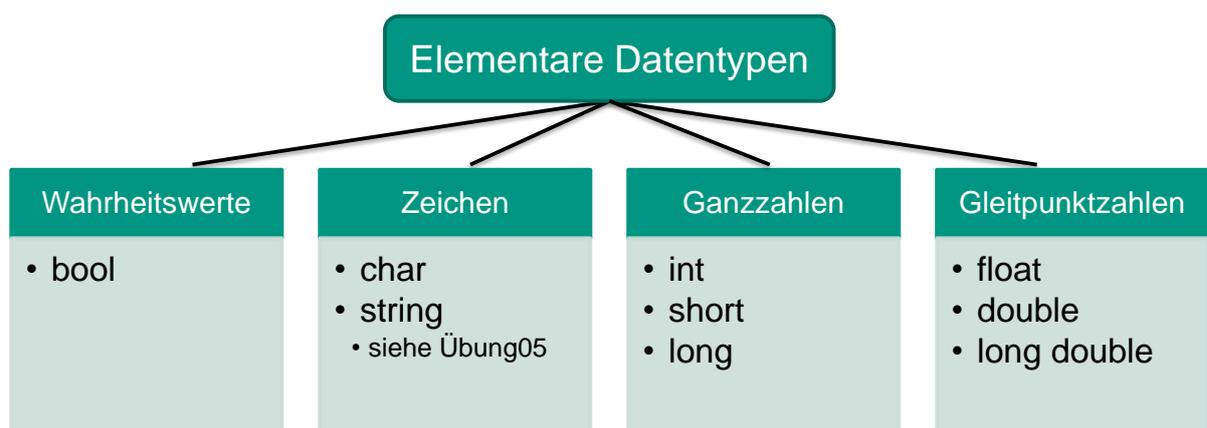
- Problemstellung
 - Wie kann ich temporär Daten speichern?
 - Wie kann ich Eingaben des Benutzers temporär speichern?

- Lösung: *Variablen*
 - Unterscheiden sich nach Typ (Festpunkt, Gleitpunkt, Char, String, ...)
 - Unterscheiden sich in ihren Verwendungsmöglichkeiten

- Abstraktion
 - Verschiedene Schubladen
 - Unterschiedliche Form
 - Unterschiedliche Größe



Variablentypen



- Ganzzahlen sind als **signed** (mit Vorzeichen) oder **unsigned** (ohne Vorzeichen) möglich



Variablentypen

Typ	Speicherplatz	Wertebereich
Variablen für Wahrheitswerte		
bool	1 Bit	true oder false
Variablen für Zeichen		
char	1 Byte	-128 bis +127 bzw. 0 bis 255
Variablen für Ganzzahlen		
int	2 Byte bzw. 4 Byte	-32768 bis +32767 bzw. -2147483648 bis +2147483647
short	2 Byte	-32768 bis +32767
long	4 Byte	-2147483648 bis +2147483647
Variablen für Gleitpunktzahlen		
float	4 Byte	3.4E+38 (Genauigkeit dezimal 6 Stellen)
double	8 Byte	1.7E+308 (Genauigkeit dezimal 15 Stellen)

 Achtung, teilweise vom Compiler abhängig!

Variablen Anlegen & Zuweisen

- Anlegen: `typ name1 [, name2, ...];`
 - Beispiel: `int zahl_x;`
 - Verwenden Sie aussagekräftige Namen

- Zuweisung eines Wertes mit „=" Zeichen
 - Beispiel: `zahl_x = 5;`
`zahl_x = zahl_x + 1;`
 - Die rechte Seite wird der linken Seite zugewiesen
 - Variable repräsentiert danach diesen Wert

- Anlegen und Zuweisen in einem Schritt
 - Beispiel: `int zahl_y = 5;`

 Nichtinitialisierte lokale Variablen haben einen zufälligen Wert

Literale / Konstanten

■ Literale

- Feste Werte → unveränderlich im Programmcode hinterlegt

■ Beispiel: `cout << "Ich lerne C++" << endl;`

- gibt "Ich lerne C++" aus / "Ich lerne C++" ist das Literal

■ Konstanten

- Variablen dessen Werte nicht mehr verändert werden können

■ Festlegen des Wertes beim Anlegen

■ Vorteil: Festlegen von Werten an zentraler Stelle

■ Schlüsselwort: `const`

■ Beispiel: `const int GESCHWINDIGKEIT = 100;`

- `GESCHWINDIGKEIT` kann als normale Variable verwendet werden, ist allerdings unveränderbar

Eingabe von der Konsole lesen und speichern

■ Einlesen von Benutzereingaben von der Konsole

■ Beispiel:

```
int zahl;
cin.sync();
cin.clear();
cin >> zahl;
```

■ Vorbereiten von Eingaben

■ `cin.sync();` ————— Puffer leeren

■ `cin.clear();` ————— Fehlerflags löschen

■ Eingabefunktion

■ `cin = console input`

■ `>>` = Zeichen von der Konsole in die Variable schieben

Zwischenübung02: Variablen

Finden Sie die Fehler im folgenden Programmcode und bestimmen Sie die Ausgabe.



```
#include "iostream"

//Zwischenübung zu Variablen
int main()
{
    int zahl_x; zahl_y = 5;
    float ergebnis = 123.45;
    char buchstabe = 'A';

    cout << Zahl_x << " " << Zahl_y << endl;
    cout << zahl_y + 2 " " 2 * ergebnis << endl;

    zahl_y = zahl_y + 5;
    cout << zahl_y << " " << buchstabe << endl;

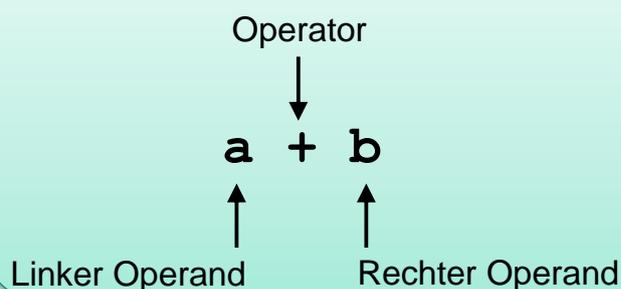
    return 0
}
```

Operatoren

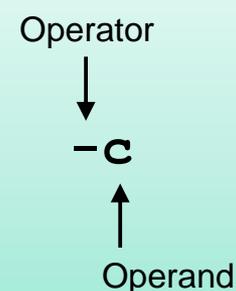
- Verarbeitung von Daten durch Operationen
- Operationen sind abhängig von der Art der Daten
 - z.B. keine Multiplikation mit nur einem Operanden
- Bei mehreren Operationen achten Sie auf die Priorität
 - Siehe C++ Kompendium für Prioritätenliste
 - Tipp: Klammern haben die höchste Priorität



Binäre Operatoren



Unäre Operatoren

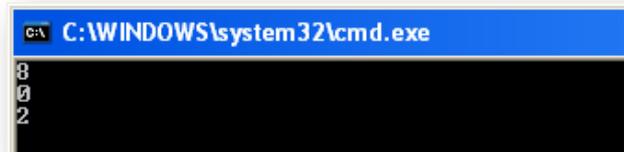


Arithmetische Operatoren

Operator	Bezeichnung
+	Addition
-	Subtraktion
*	Multiplikation
/	Division
%	Modulodivision

Beispiel:

```
int a = 5;
int b = 3;
cout << a + b << endl;
cout << a - 5 << endl;
cout << 11 % 3 << endl;
```



```
C:\WINDOWS\system32\cmd.exe
8
0
2
```



```
cout << a / 0;
```

Ungültig, da Division durch Null nicht möglich

Unäre Arithmetische Operatoren

Operator	Bezeichnung
+ -	Vorzeichenoperator (unär)
++	Inkrement-Operator
--	Dekrement-Operator

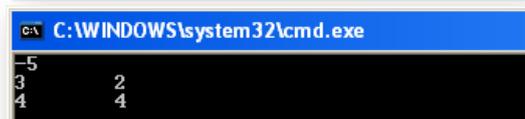
Beispiel:

```
int c = 2; int d = 5;
cout << -d << endl;
d = c++;
cout << c << "\t" << d << endl;
d = ++c;
cout << c << "\t" << d << endl;
```

Keine Veränderung von d

c wird erst d zugewiesen, danach wird c inkrementiert

c wird erst inkrementiert und dann d zugewiesen



```
C:\WINDOWS\system32\cmd.exe
-5
3 4
2 4
```



```
cout << ++n << ++n;
```

Ungültig, da mehrere Inkrement- bzw. Dekrement Operationen mit der gleichen Variablen in einem Befehl nicht definiert sind

Operatoren für binäre Operationen

Operator	Bezeichnung
~	NICHT
&	UND
^	ODER, exklusives-ODER
<< >>	Links-Shift, Rechts-Shift

unsigned int a, b, c;	Bitmuster
a = 5;	00.....00000101
b = 12;	00.....00001100
c = ~a;	11.....11111010
c = a & b;	00.....00000100
c = a b;	00.....00001101
c = a ^ b;	00.....00001001
c = b << 3;	00.....01100000
c = b >> 2;	00.....00000011

Zuweisungsoperatoren

Operator	Bezeichnung
=	Einfache Zuweisung
op=	Zusammengesetzte Zuweisung (op) ist ein arithmetischer Operator

Beispiel:

```

int a = 2; int b = 3;
double c;
a = b + 3;
cout << a << endl;
b *= a;
cout << b << endl;

! c = 5 / 2;
  cout << c << endl;
  c = 5.0 / 2.0;
  cout << c << endl;

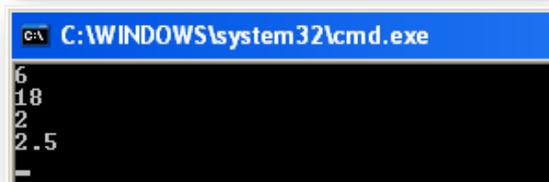
```

a wird verändert, b nicht

äquivalent `b = b * a;`

5 und 2 werden als `int` interpretiert

Interpretation als `double`



```

C:\WINDOWS\system32\cmd.exe
6
18
2
2.5
_

```

Operatorenpriorität

Priorität	Operator	Assoziativität
hoch ↑ ↓ niedrig	++ -- (Postfix)	
	++ -- (Präfix)	
	+ - (Vorzeichen)	
	! ~ (Nicht)	
	* / %	von links
	+ (Addition) , - (Subtraktion)	von links
	>> << (Rechts-, Links-Shift)	von links
	& (bitweises UND)	von links
	^ (bitweises exklusiv ODER)	von links
	(bitweises ODER)	von links
niedrig	Zuweisungsoperatoren: = += -= *= /= = <<= ...	von rechts

Assoziativität beschreibt die Reihenfolge der Auswertung

Zwischenübung03: Operatoren

```
#include <iostream>
using namespace std;
```

```
//Zwischenübung zu Operatoren
```

```
int main()
{
    int zahl_x = 12, zahl_y = -2;

    cout << 10 / 3 << " " << zahl_x % 3 << endl;
    cout << 3 + 4 % 5 << " " << 3 * 4 % 5 << endl;

    zahl_x = -4 * zahl_y++ - 6 % 4;
    cout << zahl_x << " " << zahl_y << endl;

    zahl_x = zahl_x << 2;
    zahl_y = zahl_x & zahl_y;
    cout << zahl_x << " " << zahl_y << endl;
    return 0;
}
```

Bestimmen Sie die
Ausgabe des folgenden
Programms?



Referenz & Ausblick

- Kompendium: Kapitel 1 - Erste Schritte & Hello World!
Kapitel 2 - Variablen, Zeiger und Arrays (bis inkl. 2.8)
- Tutorium: Aufgabe 1 - 4
- Wie erstellt man 100 Variablen gleichzeitig?
- Wie kann ich in einem Programm Entscheidungen treffen?
- Wie kann ich etwas 100 mal wiederholen lassen, es aber trotzdem nur einmal schreiben?
- ...

Vielen Dank für Ihre Aufmerksamkeit



Fragen?

Tobias Schwalb & Timo Sandmann & Stephan Werner
Karlsruher Institut für Technologie (KIT) – ITIV

tobias.schwalb@kit.edu

timo.sandmann@kit.edu

stephan.werner@kit.edu