

Einführung

Einführung in C im Modul
Imperative Programmierung

Wer bin ich?

M.Sc. Sven Kluge

Mitarbeiter am Lehrstuhl für Visual Computing

Email: sven.kluge@uni-rostock.de

Tel: 0381/498-7484

Wer sind Sie?

Wer sind Sie?

Wer hatte Informatik Leistungskurs/Hauptfach o.ä. in der Schule?

Wer sind Sie?

Wer nutzt Linux?

Wer sind Sie?

Wer hat Vorkenntnisse in der Programmierung?

Wer sind Sie?

Wer hat schon mit der
Kommandozeile/Shell (Windows, Linux) gearbeitet?

Wer sind Sie?

Wer hat schon Versionskontrollsysteme (SVN, git)
genutzt?

Einführung

1. Organisatorisches
2. Anmelden, Verzeichnisse, Vorkenntnisse
3. Starten und Nutzen von MinGW
4. „Hello World“ in C
5. (Lösungen)
6. Laufzeitfehler
(Der Debugger GDB und Fehlersuche)

1. Organisatorisches

- Programmiersprache C
 - C98-Standard
- Hausaufgaben
 - In Gruppen von 2-4 Studenten
 - 50% notwendig zum bestehen des Moduls
 - Anmeldung der Gruppen bis zum Ende der Woche in StudIP

2. Anmeldung und Verzeichnisse

Sie sollten über ein Nutzerkennzeichen des IT- und Medienzentrums (ITMZ) der Universität Rostock verfügen und sich möglichst damit einloggen.

Aufbau der Nutzerkennzeichens:

(1.Buchstabe Vorname+1.Buchstabe Nachname+3 oder 4 Ziffern)

Beispiel: sk1396 – Sven Kluge

Anmeldung über:

- PC-Pool Raum 310
- <https://www.itmz.uni-rostock.de/onlinedienste/anwendungsserver-des-itmz/anwendungsserver-des-itmz/>
- <https://cloud.uni-rostock.de/uniComp/>

2. Anmeldung und Verzeichnisse

Über den Laufwerksbuchstaben R: steht Ihnen ein individueller Arbeitsbereich unabhängig vom genutzten Rechner zur Verfügung.

Legen Sie dort für das Modul Imperative Programmierung ein Verzeichnis an!

Beispiel R:/Imp2019

(Hinweis: **Vermeiden Sie Leerzeichen und Sonderzeichen!**)

2. StudIP

- <https://studip.uni-rostock.de/>

Anmeldung mit ITMZ-Kürzel

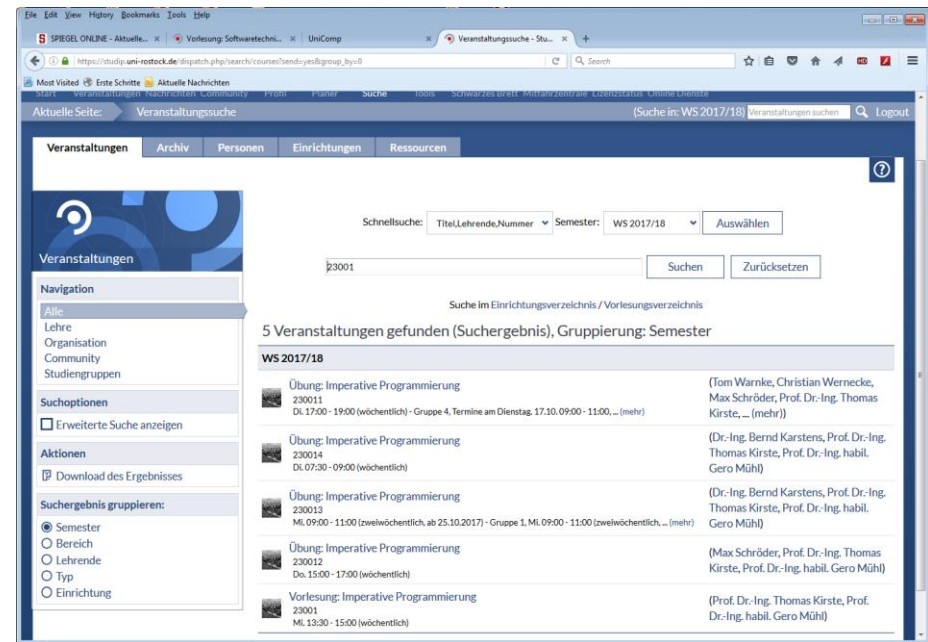
Suche nach Veranstaltung

230014 Übung Lehramt

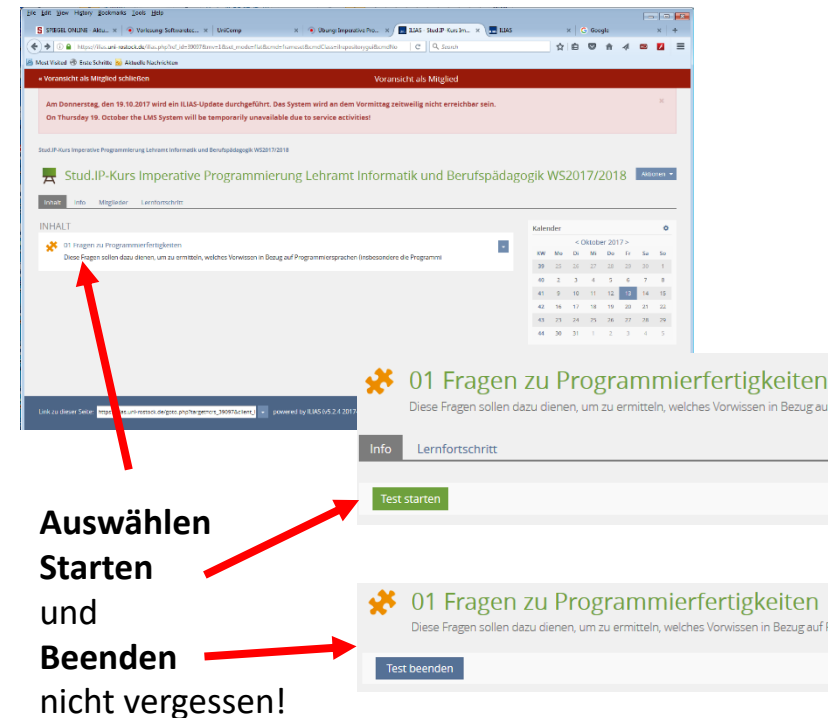
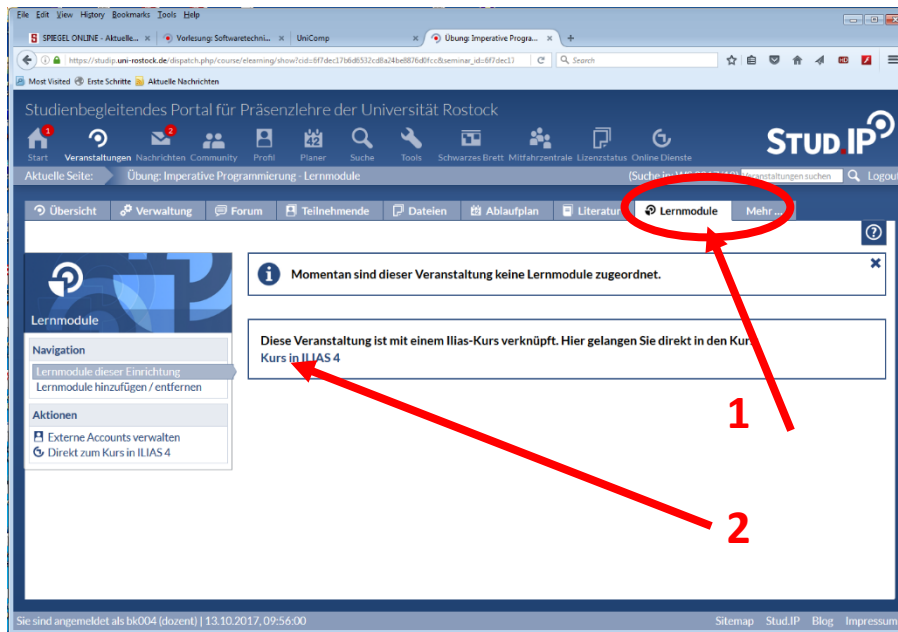
230012 Übung ITTI/MIT

(23001 –Vorlesung)

Entsprechende Veranstaltung
wählen und beitreten.



2. Fragebogen (ca. 10-15min)



3. Arbeit im Labor

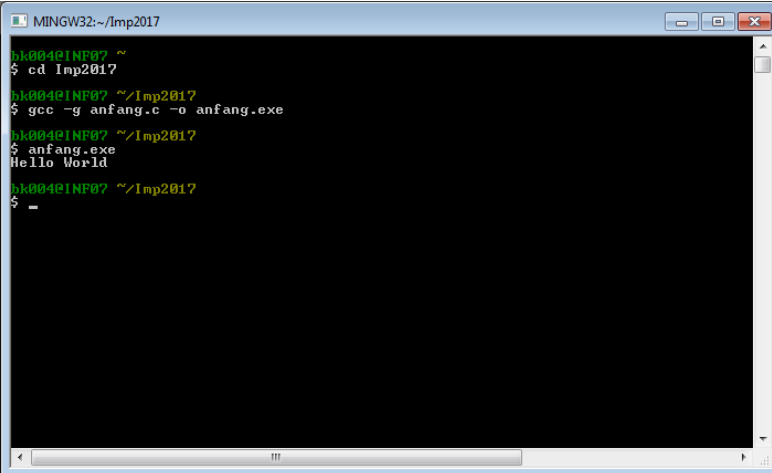
Geben sie in Notepad ++ folgendes Programm ein:

```
#include <stdio.h>
main() {
    printf("Hello World \n");
}
```

Speichern Sie dieses in Ihrem Verzeichnis (Imp2019) unter dem Namen anfang.c

Anschließend geben Sie im MinGW Shell - Fenster ein:

```
cd Imp2019
gcc -g anfang.c -o anfang.exe
anfang.exe
```



The screenshot shows a MinGW32 shell window with the following commands and output:

```
hkh004@IMP07 ~
$ cd Imp2017
hkh004@IMP07 ~/Imp2017
$ gcc -g anfang.c -o anfang.exe
hkh004@IMP07 ~/Imp2017
$ anfang.exe
Hello World
hkh004@IMP07 ~/Imp2017
$ _
```

3. Arbeiten im Labor

Programm:

```
#include <stdio.h>
main() {
    printf("Hello World \n");
}
```

Versuchen sie sich durch Fehler (Weglassen von einzelnen Zeichen (Semikolon, Klammern, Anführungszeichen) und Groß- und Kleinschreibungen von Buchstaben im Programm sich mit Fehlerausschriften vertraut zu machen.

4. Aufgaben

1. Geben Sie die Zeichenkette Hallo Welt im nachfolgenden Schema aus!
Nutzen Sie Formatsteueranweisungen \n und \t !

```
H       a       l       l       o
W
e
l
t
```

2. Wie können die Zahlen 9 und 16.25 linksbündig, rechtsbündig mit Mindestanzahlen von Stellen ausgegeben werden?

3. (Fortgeschrittene)

Berechne die Summe und das Maximum von 3 Zahlen und gebe es aus.
(Wir wollen die Zahlen 9, 3 und 7 nutzen, später beliebige 3 Zahlen)

4. (C – Experten):

Gebe die Summe und das Maximum von 5 Zahlen aus,
lasse beliebige Zahlen zur Eingabe zu.)

4. Hinweise

Formatsteuerzeichen

<code>\n</code>	Newline – Neue Zeile
<code>\t</code>	Tabulator – nächste Tabposition (mind 8 Leerzeichen)
<code>\“</code>	Ausgabe Hochkomma
<code>\\</code>	Ausgabe \
<code>\0</code>	Später noch wichtig Zeichenkettenendezeichen

4. Hinweise

```
printf("Die Zahlen %d und %f \n", 9, 16.25);
```

%d oder %i	Für Integerwerte
%X %x	Für Ausgabe in Hexadezimalsystem
%u	Für Integerwerte ohne Vorzeichen
%f	Floatwerte
%e	Floatwerte in Exponentialdarstellung
%c	Ausgabe von Buchstaben
%s	Ausgabe von Zeichenketten

4. Hinweise

Formatierungen links-und rechtsbündig, Mindestanzahl von Stellen

Durch Zeichen zwischen den % und der Formatwahl kann die Anzahl und Ausrichtung der Ausgabe gesteuert werden.

Linksbündig „-“(Minus) , Rechtsbündig(Standard ohne Formatierungszeichen)

Ganzzahl – Anzahl der Zeichen in der Ausgabe

Folgt der Ganzzahl ein Punkt und eine weitere Zahl, so sind es Mindestziffern

Folgt dem % ein „+“, so wird das Vorzeichen auf jeden Fall mit ausgegeben

%10d	Reserviere 10 Zeichen Platz für Ganzzahl, gebe die Zahl rechtsbündig aus.
%-10d	Reserviere 10 Zeichen Platz für Ganzzahl, gebe die Zahl linksbündig aus.
%10.4d	Reserviere 10 Zeichen Platz für Ganzzahl, gebe die Zahl mit mindestens 4 Ziffern (ggf führende Nullen) aus.
%10f	Reserviere 10 Zeichen Platz für Gleitpunktzahl, gebe die Zahl rechtsbündig aus
%-10f	Reserviere 10 Zeichen Platz für Gleitpunktzahl, gebe die Zahl linksbündig aus
%10.3f	Reserviere 10 Zeichen Platz für Gleitpunktzahl, gebe die Zahl mit mindestens 3 Nachkommaziffern aus
%+d	Gebe das Vorzeichen mit aus

5. Lösungen

Summe und Maximum von 3 Zahlen.
Wir berechnen im Kopf und geben aus!

```
#include <stdio.h>
main(){
    printf(" Die Summe von 9 + 3 + 7 ist 19, der maximale Wert ist 9\n");
}
```

Können wir das auch als Lückentext?

```
#include <stdio.h>
main(){
    printf(" Die Summe von 9 + 3 + 7 ist %d, der maximale Wert ist 9\n", 19);
}
```

Können wir das auch als Lückentext mit Rechenoperationen?

```
#include <stdio.h>
main(){
    printf(" Die Summe von 9 + 3 + 7 ist %d, der maximale Wert ist 9\n", 9+3+7);
}
```

5. Lösungen

Variablen mit Rechenoperationen?

```
#include <stdio.h>
main(){
    int a=9;
    int b=3;
    int c=7;
    int sum, max;
    sum = a + b + c;
    max = a;
    printf(" Die Summe von %d + %d + %d ist %d, der maximale Wert ist %d\n",
a,b,c,sum,max);
}
```

5. Lösungen

Verzweigungen?

```
#include <stdio.h>
main(){
    int a=9;
    int b=3;
    int c=7;
    int sum, max;
    sum = a + b + c;
    if (a > b)
        if (a < c) max = a; else max = c;
        else if (b > c) max = b; else max = c;
    printf(" Die Summe von %d + %d + %d ist %d, der maximale Wert ist %d\n",
        a,b,c,sum,max);
}
```

Bessere Alternative:

```
if (a > b) max = a; else max = b;
if (c > max) max = c;
```

5. Lösungen

Eingaben

```
#include <stdio.h>
main(){
int a,b,c;
int sum, max;
    printf("Eingabe erste Zahl\n");
    scanf("%d", &a);
    printf("Eingabe zweite Zahl\n");
    scanf("%d", &b);
    printf("Eingabe dritte Zahl\n");
    scanf("%d", &c);
    fflush(stdin);
    sum = a+b+c;
    if (a>b) max = a; else max = b;
    if (c>max) max=c;
    printf(" Die Summe von %d + %d + %d ist %d, der maximale Wert ist %d\n",
a,b,c,sum,max);
}
```


5. Lösungen

Eingaben und Wiederholung neuer Testfälle bis a den Wert -1!

```
#include <stdio.h>
main(){
int a,b,c;
int sum, max;
    a = 9;
    while (a != -1) {
        printf("Eingabe erste Zahl\n");
        scanf("%d", &a);
        if (a == -1) break;
        printf("Eingabe zweite Zahl\n");
        scanf("%d", &b);
        printf("Eingabe dritte Zahl\n");
        scanf("%d", &c);
        fflush(stdin);
        sum = a+b+c;
        if (a>b) max = a; else max = b;
        if (c>max) max=c;
        printf(" Die Summe von %d + %d + %d ist %d, der maximale Wert ist %d\n",
a,b,c,sum,max);
    }
}
```

5. Lösungen

Ausgabe formatierter Lückentexte!

```
#include <stdio.h>
```

```
main() {
```

```
printf("H\ta\tl\tl\to\nW\ne\nl\nt\n");  
printf("Ausgabe von \"\\\"\\\n");
```

```
printf("Die Zahlen %d und %f\n", 9, 16.25);
printf("Die Zahlen %10d und %14f\n", -9, 16.25);
printf("Die Zahlen %-10d und %-14f\n", 9, 16.25);
printf("Die Zahlen %10.4d und %14.3f\n", 9, 16.25);
printf("Die Zahlen %+d und %+14.3f\n", 9, 16.25);
```

}

6. Debugging

Programm:

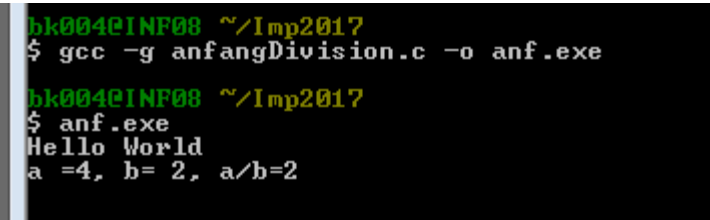
```
#include <stdio.h>
main() {
    int a;
    int b;
    a = 4;
    b = 2;
    printf("Hello World \n");
    printf("a =%d, b= %d, a/b=%d \n", a, b, a/b);
    b = 0;
    printf("a =%d, b= %d, a/b=%d \n", a, b, a/b);
}
```

6. Debugging

Programm:

```
#include <stdio.h>
main() {
    int a;
    int b;
    a = 4;
    b = 2;
    printf("Hello World \n");
    printf("a =%d, b= %d, a/b=%d \n", a, b, a/b);
    b = 0;
    printf("a =%d, b= %d, a/b=%d \n", a, b, a/b);
}
```

Fehler: Division durch 0



```
bk004@INF08 ~/Imp2017
$ gcc -g anfangDivision.c -o anf.exe
bk004@INF08 ~/Imp2017
$ anf.exe
Hello World
a =4, b= 2, a/b=2
```

6. Debugging

Das Programm an.c wird übersetzt mit:

`gcc -g anf.exe` ----- Anschließend wird der Debugger gdb gerufen:

`gdb anf.exe`

(gdb) b main Breakpoint Setze Haltepunkt auf Funktion main

(gdb) r run lasse das Programm laufen

(gdb) l list Liste den Kontext in der aktuellen Umgebung auf

(gdb) l 10 list Liste Programm ab Quelltextzeile 10

(gdb) bt backtrace Zeige die Aufrufstruktur bis zur aktuellen Stelle

(gdb) p x print variable Zeige den Inhalt von Variable x

(gdb) b main Breakpoint Setze Haltepunkt auf Funktion main

(gdb) b 9 Breakpoint Setze Haltepunkt in Zeile 9

(gdb) clear 9 Breakpoint Lösche Haltepunkt in Zeile 9

(gdb) n next Nächste Zeile abarbeiten (geht in Ergebnis der Funktion)

(gdb) s step Führe nächsten Befehl aus (geht in Funktionen)

(gdb) u until Führe alle Befehle in der Schleifen aus

(gdb) f finish Führe bis Ende der Funktion Befehle aus.

(gdb) q quit Beende Debugger

6. Debugging

```
hK004e1NF08 ~/Imp2017
$ gdb anf.exe
GNU gdb (GDB) 7.5
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-pc-mingw32".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from r:\Imp2017\anf.exe...done.
(gdb) break main
Breakpoint 1 at 0x4013be: file anfangDivision.c, line 5.
(gdb) step
The program is not being run.
(gdb) run
Starting program: r:\Imp2017\anf.exe
[New Thread 2880.0xb8c]

Breakpoint 1, main () at anfangDivision.c:5
5      a = 4;
(gdb) step
6      b = 2;
(gdb) step
7      printf("Hello World \n");
(gdb) step
Hello World
8      printf("a =%d, b= %d, a/b=%d \n", a, b, a/b);
(gdb) step
a =4, b= 2, a/b=2
9      b = 0;
(gdb) step
10     printf("a =%d, b= %d, a/b=%d \n", a, b, a/b);
(gdb) step

Program received signal SIGFPE, Arithmetic exception.
0x00401410 in main () at anfangDivision.c:10
```