



Praktikum Informatik II

Programmierung in C

Praktikum 1 – Einführung Online Praktikum

Prof. Dr. Dieter Koller

Zur C-Programmierung auf Privatrechnern

- Das Praktikum zur Informatik II findet auf Laborrechner der Hochschule statt. Die Programmerstellung kann vorzugsweise auch auf Privatrechnern erfolgen.
- Zur Programmerstellung auf den Laborrechnern der Hochschule ist sowohl Notepad++ und MS Visual Studio installiert als Visual Studio Code installiert. Diese Programme können Sie mit Hilfe der Anleitung im Moodle Kursraum auch auf Ihren Privatrechnern installieren (siehe z.B. [MSVS-Compiler](#)).
- Eine elegante Alternative besteht darin, mit Hilfe Cloud-basierter Services nur mit Hilfe eines Web-Browsers, z.B. über [repl.it](#), zu programmieren. Dazu müssen Sie nichts auf Ihrem Privatrechnern installieren.
- Auf den folgenden Seiten sind die verschiedenen Vorgehensweisen beschrieben.

Einrichtung des Rechners an der Hochschule

Für das Arbeiten an einem Rechner in einem der Rechnerräume der Hochschule gilt:

- Login mit Benutzername und Passwort.
- Persönliches Verzeichnis in Netz-Laufwerk U:/
- Neuen Ordner z.B. "U: /info2" in U: anlegen für C Programme
- Öffentliches Laufwerk R:/ (um Daten auszutauschen).
- In .profiles sind Ihre persönlichen Einstellungen gespeichert.
- „Organisieren“ -> „Ordner und Suchoptionen“ -> „Ansicht“:
 - Anzeigen von „versteckten Ordnern“ ausschalten
 - Anzeigen „Erweiterungen von bekannten Datentypen“ einschalten
- In der Pause und am Ende der Veranstaltung

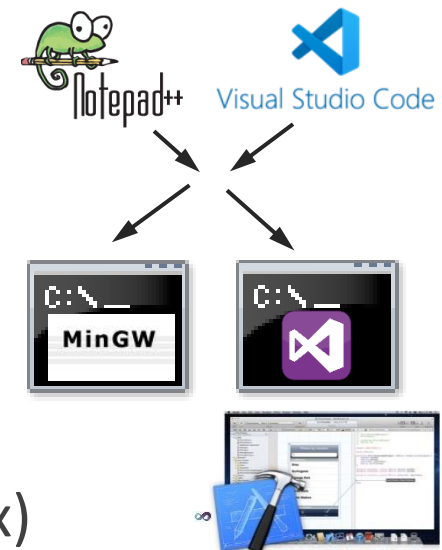
Wichtig!: Rechner durch „Windows-Taste + L“ sperren oder abmelden!

Erstellen von C-Programmen (an der Hochschule)

- Unsere Programme sind sehr kurz und klein
- Die Debug-Anforderungen sind zumindest anfangs sehr gering
- Wir wollen eine einfache und schnell erlernbare Umgebung
- Die Installation ist natürlich auch auf Privat-Notebooks möglich

Optionen

1. Microsoft Visual Studio (MSVC)
2. Notepad++ oder VS Code mit
 - a. MS C-Compiler in Windows-Konsole
 - b. GNU C-Compiler in MinGW-Konsole (Unix)
3. Xcode mit GNU C-Compiler in Apple-Konsole (Unix)








C-Quellcode erstellen mit notepad++

- Download und Installation von **notepad++** (Editor) über <http://notepad-plus-plus.org/download/>
- Syntax Highlighting fast aller gängigen Programmiersprachen
 - Unterstützung in der fehlerfreien Eingabe (Kommentare, Klammern)
 - Bessere Lesbarkeit des Quellcodes






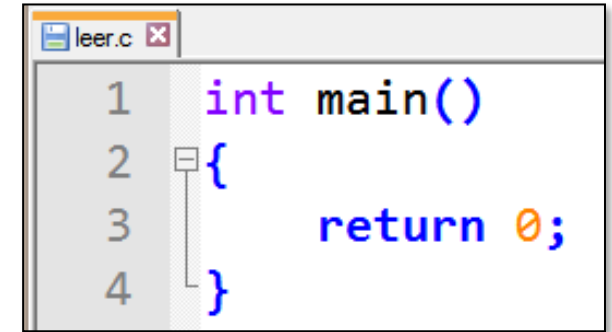
```
kap_01_2.c x
1 // Grundgerüst für C-Programm
2
3 #include <stdio.h>
4
5 main() // oder auch: int main(void)
6 {
7     printf("Hallo Welt\n");
8 }
```

Visual Studio Compiler und Notepad++ (Hochschule)

1. An der Hochschule ist **Visual Studio 2015** installiert
 - Umfangreiche und mächtige IDE inklusive Editor, Compiler, Debugger (vorinstalliert)
2. Nutzung von **notepad++** (Editor) (auf Laborrechnern installiert)
3. Nutzung **NUR** des Visual Studio Compilers über Eingabeaufforderung (Konsole)
4. Öffnen einer **Konsole** mit **Initialisierung** der Compiler-Laufzeitumgebung:
 - Zu finden unter:  "Alle Programme"
 -  "Microsoft Visual Studio 2015"
 -  "Visual Studio Tools"
 -  "Developer Eingabeaufforderung für VS 2015"
5. Navigation innerhalb der Konsole zum Verzeichnis mit den Quellcodes mit "cd".
Z.B.: Wenn Ihre Quellcodes in U:\info1 abgespeichert sind, dann Eingabe von:
"> u:" gefolgt von "> cd info1" (cd = change directory)
6. Erstellen bzw. editieren einer Quellcode-Datei mit notepad++ und abspeichern
7. Aufruf des Compilers in Konsole mit Namen der Quelldatei: 
8. Schnelles Umschalten zwischen **notepad++** Editor und Konsolenfenster mit der Tastenkombination  +  (d.h. zu Schritt 6: editieren)

Erste Schritte: Leerer Rahmen

1. Notepad++ oder VS Code starten
2. Den minimalsten (leeren) Rahmen erstellen.
3. Speichern als "leer.c" in einem Verzeichnis auf U:\
z.B. U:\Info2\. Leerzeichen vermeiden!!
4. Eine Eingabeaufforderung öffnen ( + R, dann "cmd"),
Versuch Compiler zu starten mit: "`cl /?`". Warum geht das nicht?
5. VisualStudio-Eingabeaufforderung öffnen (VS Tools), Compiler starten mit "`cl /?`".
6. Ins Übungsverzeichnis wechseln, wo die Quelldatei liegt (mit "`cd`" - Befehl)
7. Mit "`cl leer.c`" oder "`gcc -o leer.exe leer.c`" kompilieren. Fehler? Warnungen?
8. Erzeugtes Programm starten mit "`> leer.exe`". Was passiert?
8. Was macht die Hauptfunktion bei `void main(void)` statt `int main()` ?
9. Verknüpfungen von Notepad++ und VS-Eingabeaufforderung auf das Desktop legen. VS-Eingabeaufforderung Verknüpfung im Feld "Ausführen in" auf z.B. U:\Info2\ einstellen.
10. Fenster offen lassen und Umschalten zwischen Notepad++ und VS-Eingabeaufforderung mit  +  .



```
1  int main()  
2  {  
3      return 0;  
4  }
```

Erste Schritte: Einfache Textausgabe

```
#include <stdio.h>

int main()
{
    printf("Hallo Welt!");
    return 0;
}
```

1. Erste Zeile mit "`#include <stdio.h>`" einfügen.
(# = **Präprozessoranweisung**, hier zum Hinzufügen von Standard Ein- und Ausgabefunktionen definiert in `stdio.h` für die Verwendung `printf()`, `scanf()`)
2. Zeile mit "`printf("Hallo Welt");`" vor "`return 0`" hinzufügen.
3. Wieder mit z.B. "> `cl leer.c`" kompilieren. Fehler? Warnungen?
(achten Sie auf den Strichpunkt ";" am Ende jeder Anweisung)
4. Programm wieder mit "> `leer.exe`" starten. Was passiert?

Erstellen von C-Programmen mit repl.it

- Neben einem Internet-Browser benötigt man keine weiteren Programme und muss **nichts installieren**.
- Start durch Aufruf der Webseite <https://repl.it/>
- Zum Einrichten und Einstellen verfahren Sie wie in [repl.it.pdf](#) im Moodle Kursraum beschrieben. Wichtig ist, dass Sie die Standard Programmiersprache auf "C" einstellen.
- Dateien werden in der "Cloud" bei repl.it gespeichert. Dazu ist es empfehlenswert, sich einen (kostenlosen) Account einzurichten und sich anzumelden, so dass man jederzeit Zugriff darauf hat. Die erstellten Dateien sind zwar öffentlich, aber nur über einen bezahlten Account einsehbar.

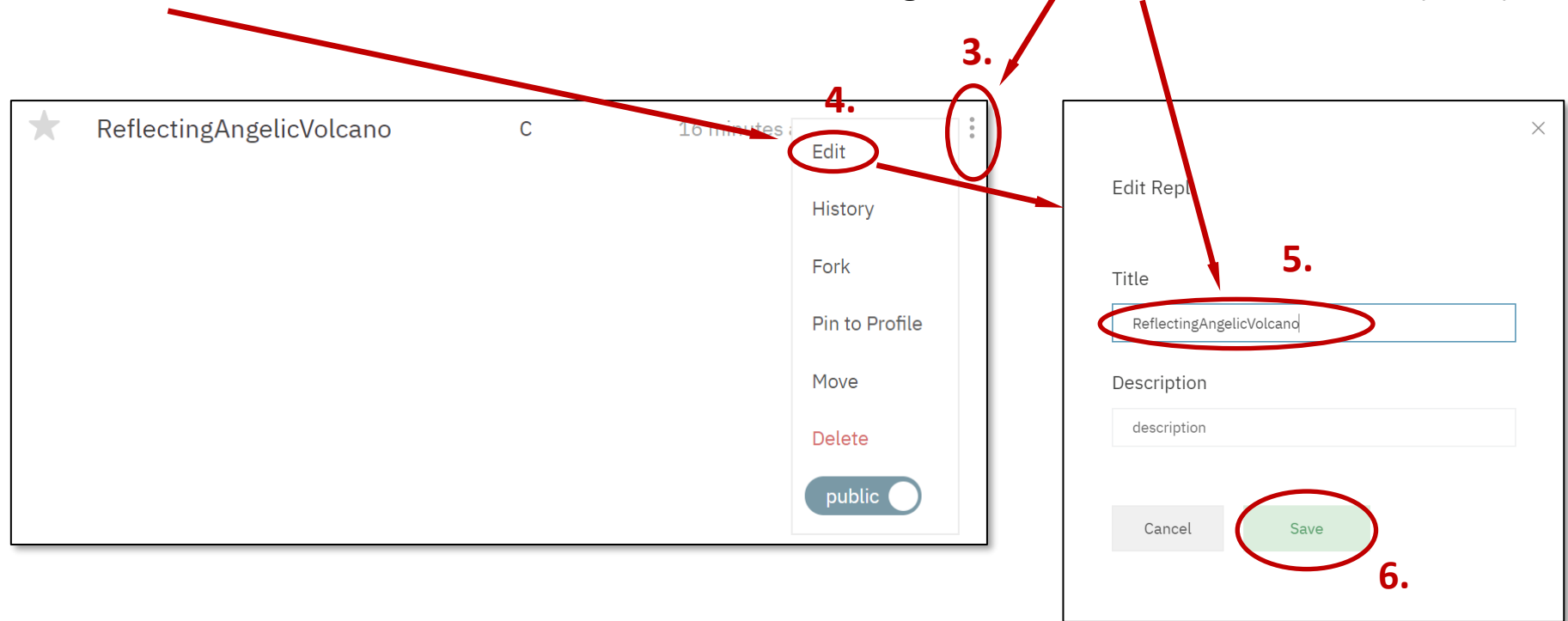
Erste Schritte: Hello World Programm mit repl.it – 1

1. Beim ersten Aufruf der Webseite <https://repl.it/> wird ein erstes Projekt (sog. Repl) mit einem Kunstnamen erstellt. Dieses Projekt enthält bereits eine `main.c` Datei mit einigen Anweisungen, ein sog. "Hello World" Programm als Template (siehe Abb. 3 in "repl.it.pdf").
2. Den Namen dieses ersten repl (hier "*ReflectingAngelicVolcano*") wollen wir umbenennen in "*HelloWorld*". Dazu selektieren wir das Menü oben links und wählen dort aus "My Repls":

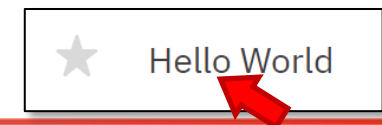


Erste Schritte: Hello World Programm mit repl.it – 2

3. Dort werden dann alle bisher erstellten Repls aufgelistet.
4. Den Namen ändern kann man durch Anwahl des Menüs (die drei vertikalen Punkte), wo wir "Edit" auswählen. Danach öffnet sich eine Dialogbox zum ändern des Namens (Title).



5. Dort löschen wir den Namen (hier "*ReflectingAngelicVolcano*") und geben ein: "*Hello World*" und bestätigen mit "Save".
6. Dann wählen wir unser Repl aus, indem wir einmal darauf klicken.



Erste Schritte: Hello World Programm mit repl.it – 3

```
main.c  saved
1  #include <stdio.h>
2
3  int main(void) {
4      printf("Hello World\n");
5      return 0;
6  }
```

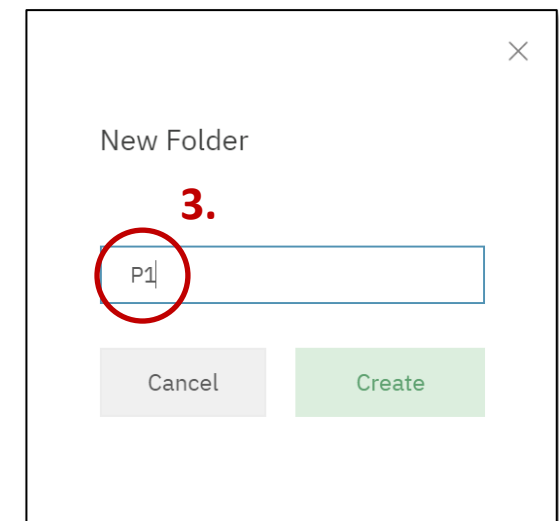
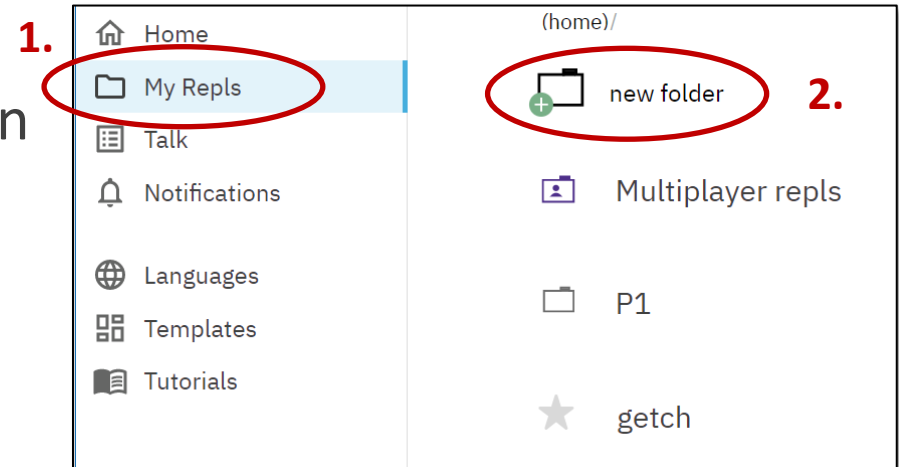


```
clang version 7.0.0-3~ubuntu0.18.04.1
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
Hello World
❖
```


1. Wählen Sie den grünen "run" Button und beobachten Sie die Ausgabe im rechten Fenster (die Eingabezeile mit clang-7 etc. braucht uns hier zunächst nicht weiter zu beschäftigen).
2. Löschen Sie den Strichpunkt ";" hinter der printf() Anweisung und wählen Sie wieder "run" aus. Was passiert jetzt im rechten Fenster?
3. Ändern Sie den Text in der printf() Anweisung zu "Hallo Welt!" und wählen Sie wieder "run" aus. Was passiert jetzt im rechten Fenster?
4. Löschen Sie die gesamte Zeile 4 und wählen Sie wieder "run" aus. Was passiert jetzt im rechten Fenster? Warum ist die Ausgabe jetzt leer?

Organisation von Dateien und Projekten in repl.it

- Legen Sie sich für jeden Praktikumstermin P1 – P7 ein separates Verzeichnis in repl.it an.
- Dazu wählen Sie im Menü ganz oben links (die drei horizontalen Striche) "My Repls" aus.
- Dort finden Sie eine Übersicht Ihres "home/" Verzeichnisses und der bisher angelegten Dateien
- Mit "new Folder" können Sie ein neues Verzeichnis anlegen, z.B. "P1" für den ersten Praktikumstermin.
- Wählen Sie dann "P1" aus und legen dort für neue Programme jeweils einen neuen repl an.



Einfache Ausgaben in C (z.B. mit Repl.it)

Erzeugen Sie mit dem Button  einen neuen Repl und nennen Sie ihn "Gruss". Modifizieren Sie die `main.c` Datei so, dass das Programm Sie begrüßt!

```
// Begrüssung

#include <stdio.h>

int main()
{
    printf("Hallo <username>!\n");
    return 0;
}
```

- Was macht der Befehl: `#include <stdio.h>` ? fügt dem Programm die header-Datei `stdio.h` hinzu, die standard ein- und ausgabe Funktionen enthält
- Wie sieht das Programm aus, wenn nach `"Hallo <username>!\n"` noch weitere Ausgaben stehen sollen? Z.Bsp.: `"Wie geht es?"`.

a) in der gleichen Zeile `printf("Hallo <username>! Wie geht es?");`

b) in der nächsten Zeile `printf("Hallo <username>! \n Wie geht es?");`

Erste Schritte: Kommentare

Die "main.c" – Datei mit Kommentaren versehen:

Wo darf welcher Kommentar stehen?

1. Vor die oberste Zeile einen einzeiligen Kommentar z.B.: `// hier Kommentar`

2. Geht auch: `int main (// das ist die Hauptfunktion) ?`

3. Mehrzeiliger Kommentar, z.B.: `/* hier
steht ein
Kommentar */`

4. Gehen auch geschachtelte Kommentare wie z.B.: `/* hier
Ja, allerdings ist der Kommentar innen nur ein Teil des Kommentars, nicht ein eigener /* und hier */
steht ein
Kommentar */ ?`

5. geht: `int main (/* Hauptfunktion */) ?` ja

6. Wann und wo benutzt man einzeilige Kommentare? Am Zeilen ende um Variablen zu beschreiben oder so

7. Wann und wo benutzt man mehrzeilige Kommentare? Um das Programm im großen Ganzen zu beschreiben oder im Code direkt, nicht am Ende der Zeile

Aufgabe 1: Verständnisfragen

Ein #-Zeichen am Zeilenanfang bedeutet

Präprozessoranweisung

Die Programmausführung beginnt

☐

a) mit der ersten include-Anweisung

☒

b) mit der ersten Anweisung in main()

☐

c) mit der ersten Funktion die im Quellcode steht

Welche Funktion benutzen wir zur Ausgabe auf dem Bildschirm (Standardausgabe)?

printf("");

Diese Funktion ist in der Standardbibliothek

stdio.h

definiert.

Was bewirkt des Zeichen '\n' am Ende des Ausgabestrings?

Zeilenumbruch

Warum benötige ich die Anweisung "return 0;" ?

Ausgabe der main funktion, einfach nur 0

Welche Funktion muss in jedem C-Programm enthalten sein?

main(void)

Aufgabe 2: Finde die Fehler in diesen Programmen?

Aufgabe a)

```
#include <stdio.h>
int main()
{
    // berühmtes Anfängerprogramm
    printf("Hello World!\n");
    return 0;
}
```

Aufgabe b)

```
#include<stdio.h >
int main()
{
    /* berühmtes Anfängerprogramm */
    printf("Hello World!\n");
    return 0;
}
```

Aufgabe 2: Finde die Fehler in diesen Programmen?

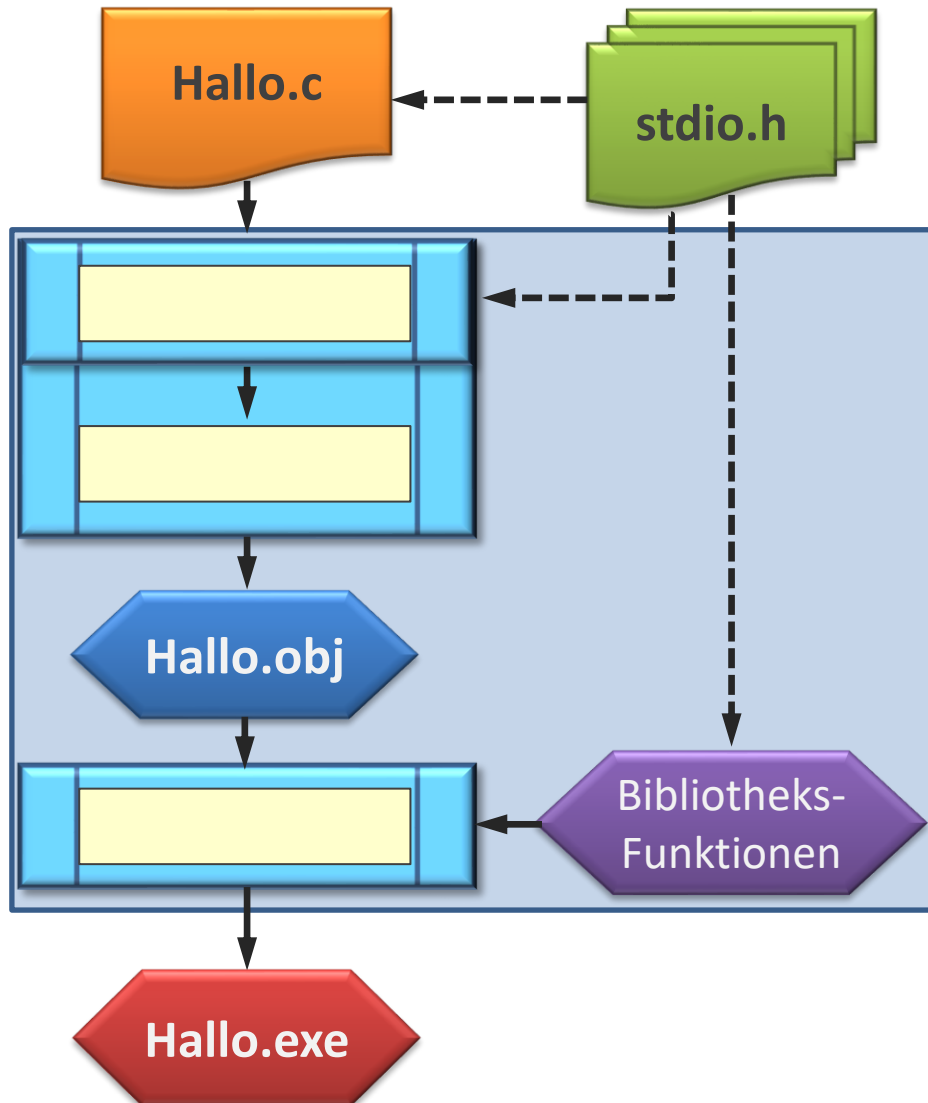
Aufgabe c)

```
#include <stdio.h>
int main()
{
    // berühmtes Anfängerprogramm
    printf("Hello World!\n");
    return 0;
}
```

Aufgabe d)

```
#include <stdio.h>
int main(){
    // berühmtes Anfängerprogramm
    printf("Hello World!\n" );
    );return 0
;}
```

Wie entsteht ein C-Programm (Ergänzen Sie)



Dateitypen

Quelldateien

*.**c** – C-Files

*.**h** – Header-Files
(Bibliotheken)

Objektdateien

*.**obj** – Object-Files

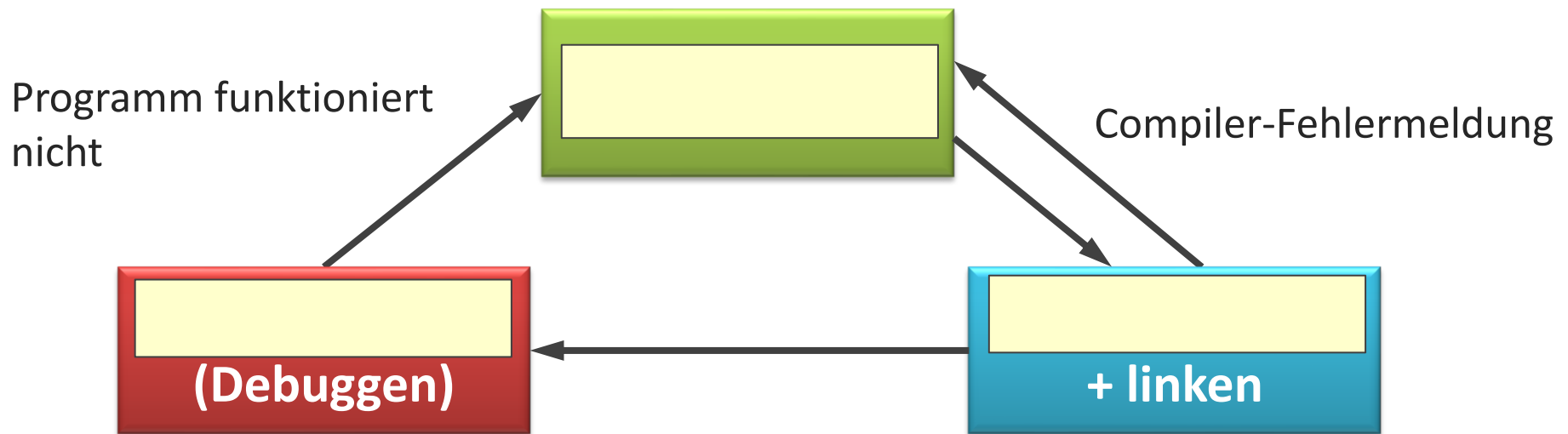
Das macht der Befehl:

>

Programme

*.**exe** – Executable

Einfacher SW Entwicklungszyklus (Ergänzen Sie)



Programmierung ist ein zyklischer Prozess

- Quellcode erstellen mit dem []
- Quellcode kompilieren mit dem [] und linken mit dem []
- Programm testen (ausführen oder testen mit dem [])

**Durch Testen kann man die Anwesenheit von []
zeigen, niemals aber deren Abwesenheit!**

Aufgabe 3: Ausgabe von Zahlenwerten in C

Schreiben Sie ein Programm, das Ihnen die Summe zweier ganzer Zahlen berechnet, z.B.:

$$7 + 12 = 19$$

Gehen Sie dabei schrittweise wie folgt vor:

- a) Geben Sie die Additionswerte sowie das Ergebnis direkt mit `printf()` aus, ohne dabei selbst die Summe zu berechnen.
Hinweis: Sie können `printf()` auch einen Ausdruck als Argument übergeben.
- b) Benutzen Sie Variablen, die Sie vorher mit
`int a, b;` definieren und dann Werte zuweisen, zum Beispiel mit
`a = 7;`
`b = 12;`
und geben Sie dann die Summe aus
- c) Benutzen Sie die Funktion `scanf()`, um vom Benutzer zur Laufzeit des Programms beliebige Werte für a und b einzulesen und geben Sie dann die Summe aus.

Aufgabe 3: Ausgabe von Zahlenwerten in C

```
4  #include <stdio.h>
5
6  int main(void){
7      int a, b;
8      printf("Bitte gib zwei Zahlen ein, die addiert werden sollen: ");
9      scanf("%d %d", &a, &b);
10     printf("%i + %i = %i", a, b, a+b);
11     getch();
12     return 0;
13 }
```