

## Konzeption zum Kurs Einführung in die Programmiersprache C

### Vorlesung 1:

Algorithmus → Programm.

- **Kapitel 0 Literatur, Software, Homepage zur Vorlesung**
- **Kapitel 1 Aufbau eines C-Programms nach ANSI-Standard** *hallo.c*
- **Kapitel 2 C-Objekte** (Alphabet, C-Variablen, C-Konstanten) *ggT.c*

### Praktikum 1:

#### Unix:

- Übungen zur Orientierung im Unix-System: Einloggen, Ausloggen, Passwort ändern, allgemeine Hinweise zum Studium im Internet, spezielle Seiten zum Kurs (Vorlesung, Aufgabenstellung), EMail zum Dateientransfer
- Auflisten des Verzeichnisinhaltes, Anlegen eines neuen Verzeichnisses *Ausdruecke.dir*, Wechsel in das neue Verzeichnis
- Editor

#### C:

Einführen der Standardfunktionen **printf()**; und der mathematischen Funktionen aus **<math.h>**. Üben an einfachen (!) Beispielen.

- Editieren, Übersetzen und Ausführen eines C-Programms *1. hallo.c*
- Einfache C-Ausdrücke,  
Verwenden von `printf()`; *2. ausdruecke\_a.c*  
Umgang mit `<math.h>` *2. ausdruecke\_b.c*

#### Hinweis: Anhang A-D, Manual Pages

\* Nur für Spezialisten

Parallel dazu Entwicklung eines Makefiles:

```
all: hallo ausdruecke_a ausdruecke_b ausdruecke_c ausdruecke_d

hallo: hallo.c
    gcc -ansi -Wall -O hallo.c -o hallo

ausdruecke_a: ausdruecke_a.c
    gcc -ansi -Wall -O ausdruecke_a.c -o ausdruecke_a -lm

ausdruecke_b: ausdruecke_b.c
    gcc -ansi -Wall -O ausdruecke_b.c -o ausdruecke_b -lm
```

---

**Vorlesung 2:**

top-Down-Analyse, switch-Anweisung

- **Kapitel 3 C-Ausdrücke**
- **Kapitel 4 C-Anweisungen**

*bandit.c*  
*rechner.c***Praktikum 2:**Einführen der Standardfunktionen `scanf()`; und der mathematischen Funktionen aus `<math.h>`. Üben an einfachen (!) Beispielen.

- Einfache C-Ausdrücke,  
Einführender Funktion `scanf()`;
- C spezifische Ausdrücke

*2. ausdruecke\_[ c / d ].c*  
*3. terms.c***Vorlesung 3:**

Sendung mit der Maus, Positionssysteme

- **Kapitel 5. Datendarstellung im Rechner** (Positionssysteme, ganze Zahlen)

**Praktikum 3:**

Anwendung von Schleifen (wann, welche, warum; was vor, nach, in der Schleife steht), Auswahlanweisungen (was vor, nach, in der Auswahlanweisung steht).

- Schleifenanweisungen
- Auswahlanweisung
- Spielprogramme

*4. quad\_[ a / b / c ].c*  
*7. quadratic.c*  
*10. raten.c 11. wuerfeln.c***Vorlesung 4:**

Maschinenzahlen.

- **Kapitel 5. Datendarstellung im Rechner** (gebrochene Zahlen)

*terms\_a.c terms\_b.c fraction.c***Praktikum 4:**

Reihenentwicklungen.

- Rechengenauigkeit
- Fehlerentwicklung bei Potenzen
- Vorbereitung zu Reihenentwicklungen:

*8. fraction.c*  
*13. potenz.c*Neuen Summand stets aus altem Summand berechnen:  $1 + x + x^2 + x^3 + x^4 + x^5$ 

summe = summand = 1;

...

summand \*= x

summe += summand

analog:

Partialsomme der Exponentialfunktion  $e^x$ :

$$\sum_{n=0}^5 \frac{(-1)^n}{(n+1) \cdot 2^n}$$

$$\sum_{i=0}^5 \frac{x^i}{i!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!}$$

**Vorlesung 5:**

Unterprogrammtechnik, Funktionen (Programm auf einer Datei),

Trennen von Anwendung und Funktionen (Programm auf mehrere Dateien).

- **6 Funktionen**

*ggT.c ggTkgV.c*  
*euklid.c.. euklid.h..ggTkgV.c kuerzen.c*

**Praktikum 5:**

Parameterübergabe bei Unterprogrammen

(lokale, globale Variable; Argumente, formale Parameter).

- Verbesserung der Exponentialfunktion aus der Vorlesung *14. exptab\_verbessert.c*
- Reihenentwicklung für die Sinusfunktion *15. sintab.c*
- \* Wurzelberechnung durch Nullstellenbestimmung *16. wurzel.c*

**Vorlesung 6:**

- **7 Rekursionen**

*erzaehle.c, fak.c, TvH.c***Praktikum 6:**

- Rekursionen

*19. Es\_war\_einmal.c**20. summe.c**21. dblFak.c*

- \* Rekursion

*22. n\_ueber\_k.c*

- **8 Konfigurationswerkzeug make**

**wird in der Vorlesung übersprungen****Vorlesung 7:**

Zeiger als drittes Grundobjekt von C, „call by value“ ist keine Einschränkung; Felder

- **9. Zeiger**
- **10 Strukturierte Datentypen - Felder**

*zeiger.c, swapping.c**lotto.c*

### **Praktikum 7:**

- Zeiger
- Felder
- Zeiger und Felder
- \* Felder mit Rekursion

*23. terms\_f.c*  
*25. scalar.c*  
*24. string.c*  
*26. permutation.c*

---

### **Vorlesung 8:**

Zeichenketten, Zeigerarithmetik; SelectSort

- **Zeichenketten**
- **Zeigerarithmetik, Selectsort**

*zahl2.c*  
*lotto.c..lottoFee.c*

### **Praktikum 8:**

- Spielprogramm

*Master Mint, Tic-Tac-Toe*

---

### **Vorlesung 9:**

Statische und dynamische Feldvereinbarung, Datenstrukturen mit Feldern und Zeigern

- **11 Dynamische Feldvereinbarung**

*vector.h, vector.c, myvector.c*  
*matrix.h, matrix.c, mymatrix.c*  
*kmd.c*

### **Praktikum 9:**

- Strukturen
- Spielprogramm

*27. brueche.c*  
*Memory*

---

### **Vorlesung 10:**

- **12 Strukturierte Datentypen - Strukturen**
- **13 Listen und Bäume**

*poltokar.c*

### **Praktikum 10:**

- Listen

*29. punktListe.c*