

# Testatblatt und Teilnahmechein für Praktikum Informatik 1

(C-Programmierung – Teil 1 für B-EI, B-MED, B-MF)

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_ Matrikelnr.: \_\_\_\_\_

Stud.-Gang: \_\_\_\_\_ Gruppe: \_\_\_\_\_ Termin (Wo-tag, Zeit): \_\_\_\_\_

Betreuer: \_\_\_\_\_ WS/SS 20\_\_\_\_ Unterschrift: \_\_\_\_\_

Im Praktikum sollen jeweils ausgewählte Übungsaufgaben aus den Kapiteln 1-25 der Aufgabensammlung und insbesondere aus den folgenden Kapiteln gelöst werden:

- 5 Ausdrücke und Operatoren
- 6 Symbolische Konstanten
- 7 Ein- und Ausgabe
- 8 Datentypumwandlungen
- 11 Die if-Anweisung
- 13 Die switch-Anweisung
- 15 Die for-Anweisung
- 16 Die while-Anweisung
- 17 Die do-while-Anweisung
- 18 Die break-Anweisung
- 22.2 Erstellen eigener Funktionen
- 22.3 Die Parameter von Funktionen
- 25.1 Eindimensionale Arrays
- 25.2 Mehrdimensionale Arrays

## Wichtig !!!

Die Übungsaufgaben sollen begleitend zur Vorlesung bearbeitet werden. An einem Praktikumstermin können **maximal 3 Lösungen** durch den Übungsbetreuer abgenommen werden.

Für Praktika besteht gemäß Studienplan Anwesenheitspflicht !

Aus vielen Kapiteln können Aufgaben unter mehreren angebotenen ausgewählt werden. Dabei müssen aus den unten angegebenen Kapiteln bzw. Unterkapiteln jeweils je nach Angabe 1 oder 2 Aufgaben gelöst werden. Die genau gewählte Aufgabennummer kann in Übg-Nr. noch ergänzt werden. Weitere angebotene Aufgaben können zusätzlich freiwillig gelöst werden.

**Bei der Lösung aller Aufgaben sind die Programmierrichtlinien der Fakultät efi zu beachten !**

Term.	Datum	Übg-Nr.	Testat	Übg-Nr.	Testat	Übg-Nr.	Testat
1.		5.2.		5.4.			
2.		5.5.		5.6.			
3.		6.		7.			
4.		7.		8.			
5.		11.		11.			
6.		13.		13.			
7.		15.		15.			
8.		16.		16.			
9.		17.		18.			
10.		22.2.		22.3.			
11.		25.1.					
12.		25.2.					

Die Aufgaben sollten bis zu folgenden Terminen im SS / WS abgegeben und testiert werden:

Termine 1 – 4: 30.4. / 15.11.; Termine 5 – 8: 31.05. / 23.12.; Termine 9 – 12: 01.07. / 15.01.

Praktikum erfolgreich abgeschlossen ☐

Datum, Unterschrift des Übungsleiters (Betreuers)

Das bestätigte Testatblatt ist möglichst am letzten Praktikumstermin, **in jedem Fall aber noch vor dem Beginn des Prüfungszeitraums**, beim Betreuer abzugeben, um rechtzeitig die erfolgreiche Teilnahme ans Studienbüro zu melden. **Verspätet abgegebene Testatblätter können im aktuellen Semester nicht mehr berücksichtigt werden !** Es wird empfohlen eine Kopie des bestätigten Testatblattes zu den eigenen Unterlagen zu nehmen.

## **Kurzfassung der Richtlinie zur Software-Entwicklung in der Programmiersprache C**

Diese Kurzfassung enthält, in knapper Form, nur die wichtigsten Regeln aus der Richtlinie. Die Einhaltung dieser Regeln ist für die Abnahme von Praktikumsaufgaben im Rahmen der C-Programmierung obligatorisch, sobald die entsprechenden Konstrukte bekannt sind ! Die Einhaltung aller Regeln aus der vollständigen Fassung wird zusätzlich sehr empfohlen. (In Klammern: Nummern der entsprechenden Regeln in der ausführlichen Fassung der Richtlinie).

- Eine Programm-Datei erhält einen Dateikopf und sinnvolle Kommentare. (R11, R21)
- Eine Programm-Datei erhält eine gut lesbare, "vernünftige" Formatierung. (R31-R39)
  - Einrücken nach {; Leerzeichen um binäre Operatoren und nach Komma; ...
- Variablen erhalten sinnvolle, aussagekräftige Namen.
  - Ein-buchstabige Namen sind nur für lokale Schleifen-Zähler erlaubt.
  - Variablen- und Funktionsnamen beginnen mit einem Klein-Buchstaben (z.B.: `varName`),
  - Namen von Konstanten werden GROSS geschrieben (z.B.: `#define ANZAHL 5`),
  - Typnamen beginnen mit einem Groß-Buchstaben. (R41-R47)
- Konstante Zahlen in Programm-Anweisungen und Ausdrücken sind verboten. Stattdessen sind `#define`-Konstanten oder Konstanten mit sinnvollen Namen zu definieren und zu verwenden.  
z.B.: `#define ANZAHL_TUEREN 5` oder: `const int anzahlTueren = 5;`  
(Ausnahmen: 0, 1 und allgemeingültige Konstanten wie bei Std/Tag: 24, min/Std: 60, ...). (R51)
- Definitionen neuer Datentypen erfolgen mit `typedef` und neuem Typ-Namen.  
z.B.: `typedef unsigned int UINT; typedef struct person { ... } Person;` (R61)
- Definitionen von Variablen und Funktionen stehen immer in .c-Dateien und nicht in .h-Dateien. Definitionen von Variablen erfolgen immer mit einem geringsten möglichen Geltungsbereich. (Lokale Gültigkeit; Keine programm-globalen Variablen ! „information hiding“). (R81, R82)

**Alle Programme sind mit der höchsten Warnstufe zu compilieren, i.e. mit:**

**`gcc -Wall -O2 ...`, `cl /wall /O2 ...`, o.ä.. **Alle gemeldeten Compiler Warnungen sind zu überprüfen und ggf. zu eliminieren oder zu erklären.****