


Übungsblatt 1 Meine Daten

Lernziele:

- Datenausgabe mit `printf()`; erlernen
- Steuerzeichen `\n` kennenlernen
- Erstellen eines einfachen C-Projektes üben

Legende:


 Lektüre, die vor der Übung gelesen werden muss

 Fragen/Aufgaben, die vor der Übung zu bearbeiten sind

 Aufgaben, die in der Übung zu bearbeiten sind

1.1 Vorbereitungen zur Übung

Die Vorbereitungen und Fragen sind vor dem Übungstermin zu bearbeiten. Sie sind als Hilfen gedacht, um Ihnen die Aufgaben, die an den Übungsterminen selbst zu bearbeiten sind, zu erleichtern.

 Lesen Sie zur Vorbereitung der Übung das Kapitel „Einführung in die Programmierung“, die Anleitung „Eclipse_cdt_intro_deutsch“ (beide auf Moodle) und das Kapitel 2 im Buch „C von A bis Z“ von Jürgen Wolf!

 **Einleitende Fragen:**

Wie gehen Sie vor, wenn Sie ein neues C-Projekt anlegen wollen? Welchen Project Type und welche Toolchain wählen Sie?

1. Eclipse öffnen. // 2. Im Menü: File -> New -> C Project // 3. Einen Projektnamen eingeben // 4. Empty C Project wählen // 5. CygwinGCC als Toolchain wählen

Was bedeutet `#include <stdio.h>`?

Diese Zeile bindet die Standard Input/Output-Bibliothek ein, welche die Funktionen `printf()` oder `scanf`

Was macht ein Compiler?

Ein Compiler übersetzt den Quellcode eines Programms in ausführbaren Maschinencode, den der Computer verstehen und ausführen kann.

Was müssen Sie vor dem Compilieren beachten?
(Mehrfachnennungen sind möglich)

- ☒ alte Projekte geschlossen
- ☒ hinter jeder Zeile muss ein Semikolon stehen
- ☒ keine Umlaute/Leerzeichen/Punkte im Dateinamen
- ☐ Programm muss einmal ausgeführt worden sein
- ☒ Projekt gespeichert
- ☒ richtige Toolchain ausgewählt
- ☐ wenn im Funktionskopf `main(void)` steht, muss das `return 0` weglassen werden

Wie compilieren Sie ihr Projekt?

- ☐ mit dem Disketten-Symbol
- ☒ mit dem Hammer-Symbol
- ☐ mit dem Pfeil-Symbol

 **Ein kleines Beispiel zum Aufwärmen:**

Erweitern Sie folgenden Funktionsrumpf zu einem „Hallo Welt“ Programm!

```
#include < stdio.h >

int main ( void )
{
    printf ( „Hello World! \n“);

    return 0;
}
```

1.2 Ausgabe einiger Daten

Sie wollen Ihre Daten wie Adresse, Telefonnummer, Matrikelnummer und Studiengang in jeweils einer neuen Zeile auf dem Bildschirm ausgeben lassen.



Frage:

Wie wird in der C-Funktion „printf“ ein Zeilenumbruch erreicht?

Durch ein `\n`.

❖ Schreiben Sie ein C-Programm, das die Informationen am Bildschirm ausgibt.

TIPPS:

- Legen Sie zunächst ein neues C-Projekt namens „Meine Daten“ und ein Source-File namens „meinedaten.c“ an.
- Implementieren Sie anschließend nur den Funktionsrumpf der `main`-Funktion.
- Nun können Sie die Ausgabe Ihrer Daten mittels einer `printf`-Funktion implementieren.

Es gibt zwei Wege, um die Daten auszugeben:

- a) Jede Information durch eine separate `printf`-Funktion ausgeben zu lassen
- b) Alle Informationen durch eine einzige `printf`-Funktion ausgeben zu lassen

Ihnen ist freigestellt, für welchen Weg Sie sich entscheiden.

1.3 Ausgabe der Daten durch Zahlenargumente

Ziel ist es, die Zahlen in Ihren Daten als Zahlargument durch eine `printf`-Funktion auszugeben.

Beispiel:

```
printf("Es ist %d Uhr", 9);
```

Hier wird, statt die Zahl direkt in den Text einzugeben, diese durch einen so genannten Formatbuchstaben ersetzt und die Zahl erst nach dem Anführungszeichenende genannt. Diese Schreibweise ist vor allem später bei der Verwendung von Variablen von Bedeutung.

☞ Sehen Sie sich die Beispieldatei „einfache__dezimalformatierungsausgabe“ an! Lesen Sie zusätzlich das Kapitel 3 „Ein- und Ausgabe“ unter folgendem Link nach:

<http://www.tutorials.at/c/c-tutorial.html>

❖ Ändern Sie Ihr Programm so ab, dass Hausnummer, Postleitzahl, Telefonnummer und Matrikelnummer durch ein Zahlargument ausgegeben werden! Hinweis: Für ganze Zahlen ist der Formatbuchstabe im Normalfall `%d`!

- a) Schreiben Sie jede Ausgabezeile in eine separate `printf`-Funktion
- b) Geben Sie die Hausnummer, Telefonnummer, Postleitzahl als Zahlargument durch `printf` aus.