

# Lehrveranstaltung "Grundlagen der Informatik"

## Übungsblatt 1

### Hinweise:

Dieses Übungsblatt ist zur Zulassung zu der Klausur erfolgreich zu bearbeiten ("*Erfolgreich*" bedeutet: Keine Programmabstürze bzw. Endlosschleifen, Aufgabenstellung einschl. der Nebenbedingungen müssen eingehalten sowie Kommentierung und Einrückung korrekt sein! Compilerwarnungen sollen möglichst vermieden werden.).

Die Aufgaben werden überwiegend in den Übungszeiten bearbeitet und dort auch abgegeben. Allerdings genügt die Zeit hierfür unter Umständen nicht, so dass Sie auch außerhalb dieser Zeiten die Aufgaben bearbeiten müssen. Der Abgabetermin für diese Aufgabe ist der **11. April 2025**.

---

**Aufgabe:** Ziel der ersten Übung ist das Kennenlernen aller Arbeitsmittel, die in diesem Semester benötigt werden: Linux, Terminalfenster, Texteditor vi, C-Compiler gcc, Fehlermeldungen des Compilers.

Im ersten Teil werden wir alles Schritt für Schritt gemeinsam machen. Schauen Sie sich dazu das Video zur Übung 1 an.

In einem Terminalfenster unter Linux werden wir die wichtigsten Befehle kennenlernen und ausprobieren:

- <code>ls</code> bzw. <code>dir</code>	Inhaltsverzeichnis anzeigen
- <code>cd</code>	Verzeichnis wechseln (change directory)
- <code>mkdir</code>	Unterverzeichnis erstellen (make directory)
- <code>rm</code>	Datei(en) löschen (remove)
- <code>rmdir</code>	Unterverzeichnis löschen (remove directory)

Dann werden wir die ersten Dateien erstellen und bearbeiten mit dem Texteditor vi (siehe auch Skript "Texteditor vi" im Downloadbereich).

Das erste, selbst erstellte C-Programm werden wir dann gemeinsam compilieren, linken und starten (sofern keine Fehler aufgetreten sind). Dabei lernen wir die wichtigsten Optionen des Compilers kennen:

- <code>gcc -c CDatei</code>	Compiliert die angegebene Datei <i>CDatei</i> in die gleichnamige Objektdatei (Dateiendung <i>.o</i> ).
- <code>gcc -o PDatei ODatei</code>	Linkt die angegebene Objektdatei <i>ODatei</i> zum Programm (ausführbare Datei) <i>PDatei</i> .

Im zweiten Teil der Übung werden Sie eigenständig das Programm `ueb01.c` compilieren, linken und starten.

Legen Sie dazu erst einmal ein Verzeichnis `ueb01` an. Speichern Sie dann den Quelltext aus dem Übungsbereich in dieses Verzeichnis, in dem Sie mit der rechten Maustaste auf den Link in der Übungsübersicht

klicken und in dem Kontextmenü den Menüpunkt '*Ziel speichern unter*' ('*Save Target as*' falls Sie eine englische Version haben) wählen. Compilieren Sie nun die Quelltextdatei und linken Sie anschließend die daraus resultierende Objektdatei zur ausführbaren Datei ueb01.

Dummerweise sind in dem Quelltext einige Fehler enthalten. Anhand der Fehlermeldungen des Compilers sollen Sie die Fehler ermitteln und beseitigen. Es sind aber auch Fehler enthalten, die vom Compiler nicht gefunden werden (sogenannte logische Fehler). Auf welchem Weg können Sie diese finden? Richtig: Durch genaues Hinschauen, Verstehen der Begrifflichkeiten (z.B. was ist Netto und Brutto?) und Nachrechnen der Ergebnisse.

Wenn Sie alle Fehler gefunden und korrigiert haben, zeigen Sie den korrigierten Quelltext vor bzw. laden Sie die korrigierte Quelltextdatei in Moodle hoch. Experimentieren Sie ruhig noch mit dem Compiler, um ihn besser kennenzulernen, z.B. in dem Sie andere Fehler einsetzen und gucken, welche Fehlermeldungen dazu erzeugt werden. Ab der nächsten Übung werden diese Kenntnisse vorausgesetzt!

**Quellcode:** #include <stdio.h>

```
int main()
{
    double Preis1 = 9.99, Preis2 = 19,99, Preis3 = 24.49;
    double Netto, Mehrwert, Brutto;

    printf("\nDieses Programm berechnet aus 3 Werten (z.B. ");
    printf("Artikelpreise) die Nettosumme, daraus die Mehrwertsteuer ")
    printf("und schliesslich den Bruttobetrag.\n");
    printf("Die drei Werte: %.2f, &.2f und %.2f\n", Preis1, Preis2, Preis3);
    Netto = Preis1 + Preis2 + Preis3;
    printf("Nettosumme      : %.2f\n", Netto);
    Mehrwert = Netto x 0.19;
    printf("Mehrwertsteuer: %.2f\n", Mehrwert);
    Brutto = Netto - Mehrwert;
    printf("Bruttobetrag   : %.2f\n", Brutto);

    return 0;
}
```