

# Das erste Programm

## Allgemeine Anmerkungen zur Aufgabenstellung:

- Beantworten Sie die Kontrollfragen in TUWEL bevor Sie mit der Übung beginnen.
- Stellen Sie während der Übung laufend sicher, dass Ihr Programm kompilierbar ist und richtig funktioniert!
- Die Abgabe erfolgt durch Uploaden Ihres vollständigen `Code::Blocks`-Projektordners als zip-Datei (`Matrikel-Nr_UE1.zip`) in TUWEL. Anschließend melden Sie sich für das Abgabegespräch in TUWEL an. Dieses erfolgt über die Telekonferenz-App Zoom ([www.zoom.us](http://www.zoom.us)). Stellen Sie sicher, dass diese App rechtzeitig vor Beginn des Abgabegesprächs auf Ihrem PC installiert ist und dass Sie Ihr aktuelles Übungsprogramm in `Code::Blocks` für die Abgabe geöffnet haben. Testen Sie bitte auch die korrekte Funktionalität Ihrer Videokamera, da Sie zu Beginn Ihres Abgabegesprächs Ihren Studentenausweis herzeigen müssen und auch während des Gesprächs eine aktive Videoverbindung erforderlich ist.
- Beachten Sie, dass Upload und Terminauswahl für das Abgabegespräch erst möglich sind, wenn Sie 80% der Kontrollfragen richtig beantwortet haben. Diese müssen, genauso wie die Übung, bis spätestens **29.03.2020 23:59** durchgeführt werden.
- Inhaltliche Fragen stellen Sie bitte im zugehörigen Forum in TUWEL. Das Betreuungsteam steht Ihnen in den jeweiligen Übungswochen von Mo.-Fr. in der Zeit von 09:00-17:00 zur Verfügung. Außerhalb dieser Zeiten sind wir auch bemüht Ihre Anfragen möglichst zeitnah zu beantworten.
- Für eine positive Beurteilung Ihrer Abgabe **muss** diese kompilierbar sein, Ihr Programm fehlerfrei terminieren und die in der Angabe angeführten Funktionen mit den vorgegebenen Funktionsköpfen enthalten. Des Weiteren müssen die Funktionen hinreichend getestet sein. Sie müssen in der Lage sein, Ihr Programm beim Abgabegespräch zu erklären!

## Folgende Hilfsmittel stehen Ihnen in TUWEL zur Verfügung:

- Alle bisherigen Referenzbeispiele sowie Vorlesungsfolien
- Das Buch zur LVA "Programmieren in C" (Robert Klima, Siegfried Selberherr)
- Kurzanleitung für die Entwicklungsumgebung `Code::Blocks`

**Hinweis:** Schreiben Sie Ihr Programm selbstständig! Das Kopieren der Lösung stellt einen groben Verstoß gegen die Durchführungsbestimmungen der LVA (siehe LVA-Beschreibung im TISS) dar, welcher zu einer negativen Beurteilung der Übung führt.

## Einleitung

In dieser Übung werden Sie Ihr erstes kleines C-Programm schreiben, und es anschließend compilieren und ausführen. Sie werden die Grundlagen und Vorzüge einer Entwicklungsumgebung anhand von `Code::Blocks` kennenlernen. Eine Einführung in die Bedienung von `Code::Blocks` finden Sie in der Kurzanleitung zur Benutzung der Entwicklungsumgebung. Weiters sollen Sie in dieser Übung Ihr Programm mit dem in der Entwicklungsumgebung integrierten Debugger analysieren. Um Ihnen die späteren Übungen zu erleichtern, wird empfohlen, sich mit den Funktionen des Debuggers genau auseinanderzusetzen. Während Ihres Abgabegesprächs werden Sie unter Umständen aufgefordert, Ihr Programm Schritt für Schritt mit Hilfe des Debuggers zu erklären.

Sie benötigen für diese Übung folgende Punkte aus dem Stoff-Gebiet:

- Übersetzen eines Programms (Kapitel 3)
- Die Funktion `main` sowie einfache Ausgabeoperationen mit `printf()` (Kapitel 4)
- Variablen, Konstanten, Datentypen `double` und `long` (Kapitel 5)
- Numerische Datentypen `char`, `long`, `float`, `double` (Kapitel 6)
- Ausgabe mit Platzhaltern: `printf()` (Kapitel 7)
- Ausdrücke (Kapitel 8)



## Teil A

- ✗ Öffnen Sie die Entwicklungsumgebung `Code::Blocks` und starten Sie ein neues Projekt wie es in der Kurzanleitung der Entwicklungsumgebung beschrieben ist.
- ✗ Schreiben Sie nun ein C-Programm, welches den folgenden Text ausgibt.

```
*****  
*           Programmieren 1           *  
*                                     *  
*       Vorname, Nachname             *  
*           Matrikelnummer            *  
*           Geburtsdatum              *  
*****
```

Dabei sollen Sie Ihren Namen, Ihre Matrikelnummer und Ihr Geburtsdatum einsetzen.

- ✗ Kompilieren Sie Ihr Programm und führen Sie es aus!



## Teil B

- ✗ In der Datei `fehler.txt` befindet sich ein C-Quelltext, der mit Fehlern versehen ist. Fügen Sie den Quelltext in Ihre `main()` Funktion im Anschluss an die vorherige Ausgabe ein („Copy“ und „Paste“). Versuchen Sie anschließend das Programm zu kompilieren und dabei auftretende Fehler zu beheben. Überprüfen Sie auch die Bildschirmausgabe.

Benutzen Sie die Compiler-Fehlermeldungen um den Quelltext zu korrigieren! Beginnen Sie bei der Fehlerbehebung jeweils beim ersten gemeldeten Fehler, da oftmals weitere angezeigte Meldungen Folgefehler sein können.



## Teil C

- ✗ Der Debugger stellt ein komfortables Werkzeug zur Fehlersuche dar. Dieser ist im Besonderen dann vorteilhaft wenn `printf` Anweisungen nicht mehr ausreichend sind um Fehler zu lokalisieren. Mit ihm können Sie die Ausführung eines Programms an definierten Stellen anhalten und danach schrittweise abarbeiten. Mit dem Debugger lässt sich so beispielsweise der Inhalt von Variablen verfolgen, feststellen ob Funktionen richtig aufgerufen werden und überprüfen ob Selektionszweige oder Schleifen richtig abgearbeitet werden. Um Ihnen die Fehlersuche in nachfolgenden Übungen zu erleichtern, machen Sie sich bitte mit den Funktionen des Debuggers vertraut.
- ✗ Setzen Sie zum Beispiel einen sogenannten Haltepunkt (Breakpoint) in Ihrem Programm und starten Sie es im Debugmodus.
- ✗ Gehen Sie Ihr Programm schrittweise durch. Beobachten Sie dabei welche Programmzeile als nächstes ausgeführt wird und welche Werte die verwendeten Variablen einnehmen.