

**Ingenieurinformatik (FK 03)**  
**Übung 1**

**EDV-LABOR EINWEISUNG**

**Sicherheitsunterweisung:**

- Es gilt die Laborordnung
- In den Laboren nicht Rauchen, Essen und Trinken
- Fluchtwege aus dem Labor auf den Flur ins Treppenhaus
- Grüne Fluchtwegemarkierungen an der Flurdecke
- Feuerlöscher auf dem Flur, Feuermelder in beiden Treppenhäusern
- Im Brandfall keinen Aufzug benutzen (möglicher Stromausfall, Kaminwirkung)
- Im Brandfall die Fenster geschlossen halten
- Informationen an den Türen:
  - Verhalten im Brandfall, Rufnummern für den Notfall, erste Hilfe Ein Unfall – was ist zu tun? - Verbandskästen in den Räumen B362 , B372 , B0055 (Sekretariat)
  - Notausschalter sind in allen KCA-Laboren vorhanden

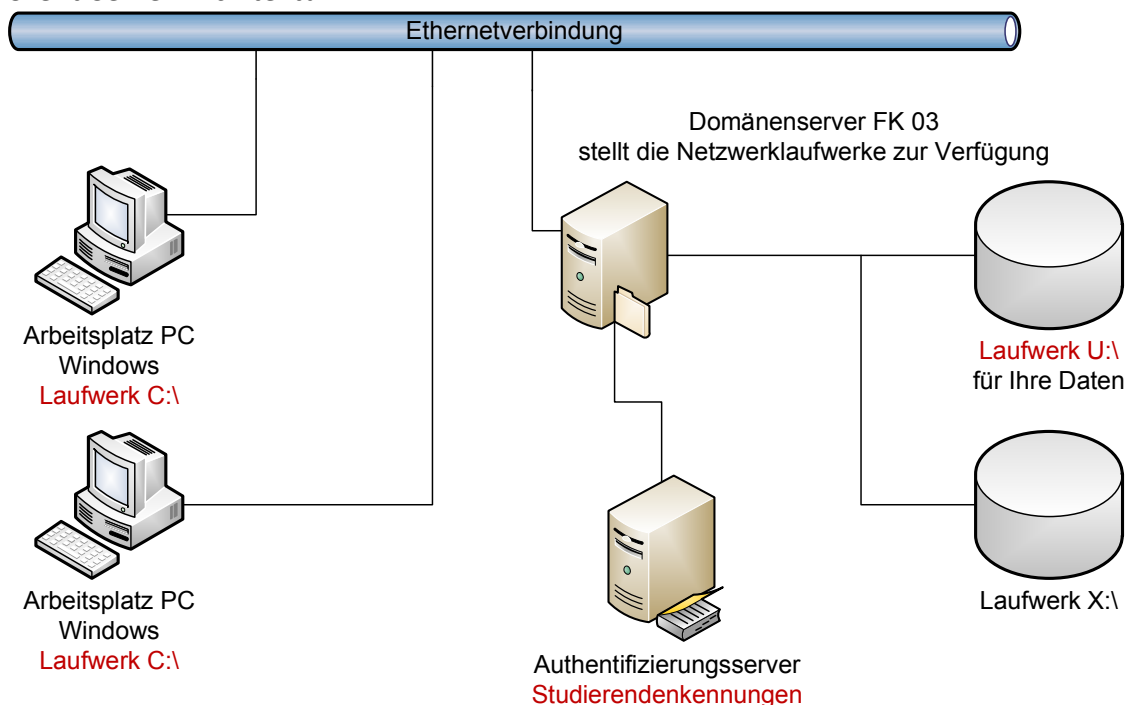
**Hard- und Softwareproblemen:**

- z. B. Login nicht möglich, Fragen zu Netzwerklaufwerken/Backups usw. wenden Sie sich an Herr Schneider, Raum B350

**EDV-LABOR AUFBAU**

Das EDV-Labor besteht aus PCs (Clients) mit Intel-Prozessoren

- Betriebssystem Microsoft Windows
- PCs sind vernetzt; mit Servern der HM und dem Internet verbunden
- Client-Server-Architektur



**Sichern Sie Ihre Arbeit immer auf **Laufwerk U:**, damit sie Ihnen auf anderen Arbeitsplätzen ebenfalls zur Verfügung steht.**

## ÜBUNG 1A (TEXTDATEIEN)

### Einführung

Texteditoren sind Programme zum Erstellen einfacher Texte, d. h. die Texte enthalten keine Grafiken und es gibt nur einfache Formatierungsbefehle (z. B. Zeilenumbruch oder neue Seite). Der Text wird im ASCII-, UTF-8 oder Unicode-Format abgespeichert.

#### Beispiele für Texteditoren:

**Edit**, **Notepad**, **SciTe** oder **TextPad**

**Beachte:** Microsoft Word ist kein Texteditor, kann aber Text im ASCII-Format (Dateityp: nur Text \*.txt) speichern. Dann gehen aber Formatierungen verloren.

### Ziel

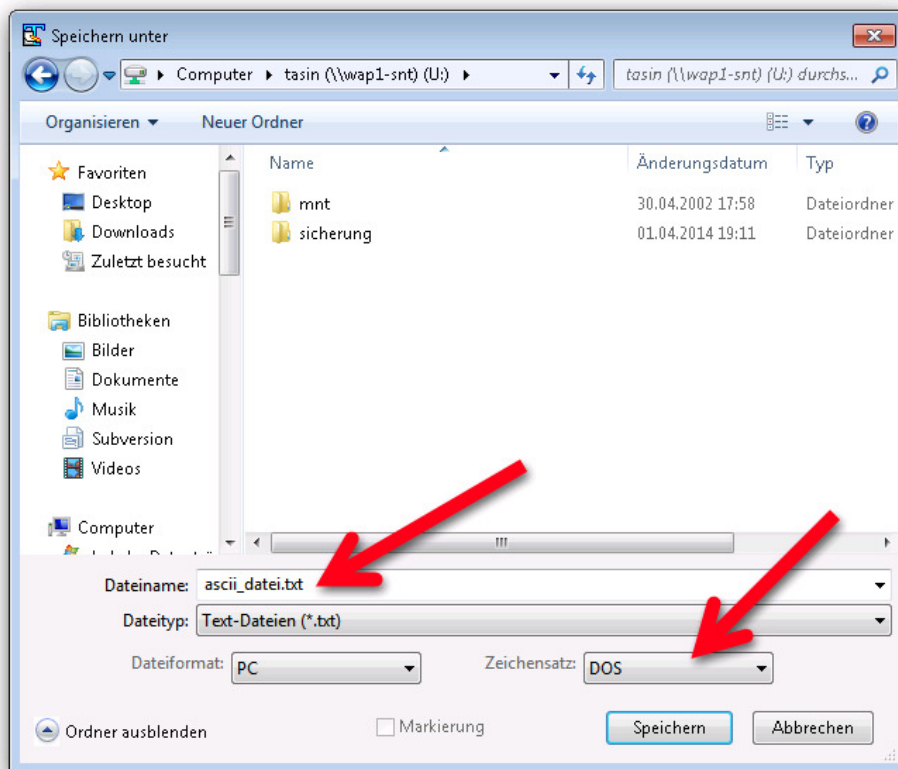
Es soll eine einfache Textdatei in den unterschiedlichen Kodierungsverfahren auf dem Benutzerlaufwerk **U**: abgelegt werden und die Datenablage in Bytes nachträglich untersucht und interpretiert werden.

### Durchführung

- Melden Sie sich mit Ihrem moodle-Account an Ihrem zugewiesenen Rechner an.
- Starten Sie das Programm **TextPad**.
- Schreiben Sie folgende Information in das Textfeld:

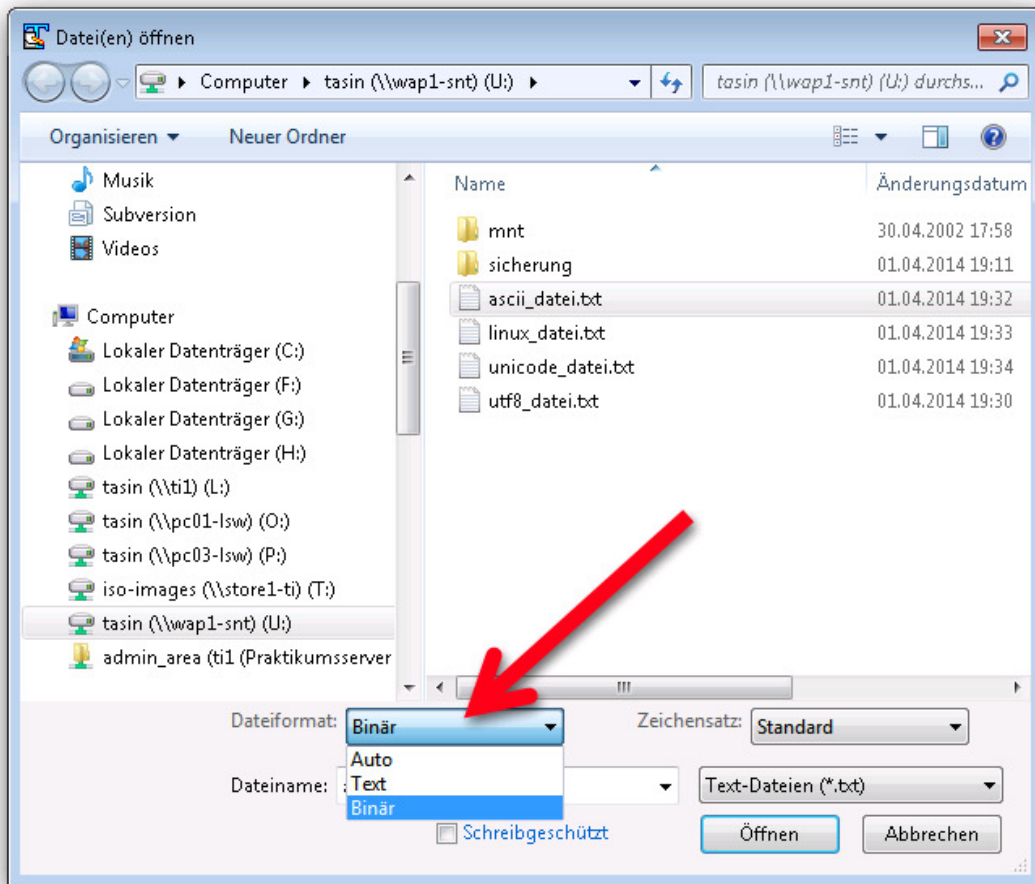
```
Meine erste Textdatei
1234
äöüÄÖÜß
```

- Sichern Sie die Datei mit dem Zeichensatz **DOS** auf Laufwerk **U**: unter dem Namen **ascii\_datei.txt**.



- Speichern Sie den gleichen Inhalt mit dem Zeichensatz **Unicode** unter **U**: **unicode\_datei.txt** und mit dem Zeichensatz **UTF-8** unter **U**: **utf8\_datei.txt**.

- Speichern Sie danach die Datei im Zeichensatz UTF-8 und dem Dateiformat **Unix** unter **U:\unix\_datei.txt**.
- Öffnen Sie nun die erstellten Dateien mit TextPad im **Binärformat**:



- **Frage A1:** Interpretieren Sie die Inhalte, stellen Sie die Gemeinsamkeiten und die gefundenen Unterschiede fest und notieren Sie sich die Ergebnisse.

## ÜBUNG 1B (ERSTES C-PROGRAMM)

### Einführung

Ein C-Programm wird im Praktikum mithilfe der Entwicklungsumgebung **QT Creator** erstellt. Eine Entwicklungsumgebung (IDE) unterstützt Sie bei der Erstellung der Quellcodedateien (Textdateien in C-Syntax) und der Umwandlung in eine lauffähige Anwendung.  
s. a. [http://pc01-lsw.ee.hm.edu/wicked/QTC\\_Projekterstellung](http://pc01-lsw.ee.hm.edu/wicked/QTC_Projekterstellung)

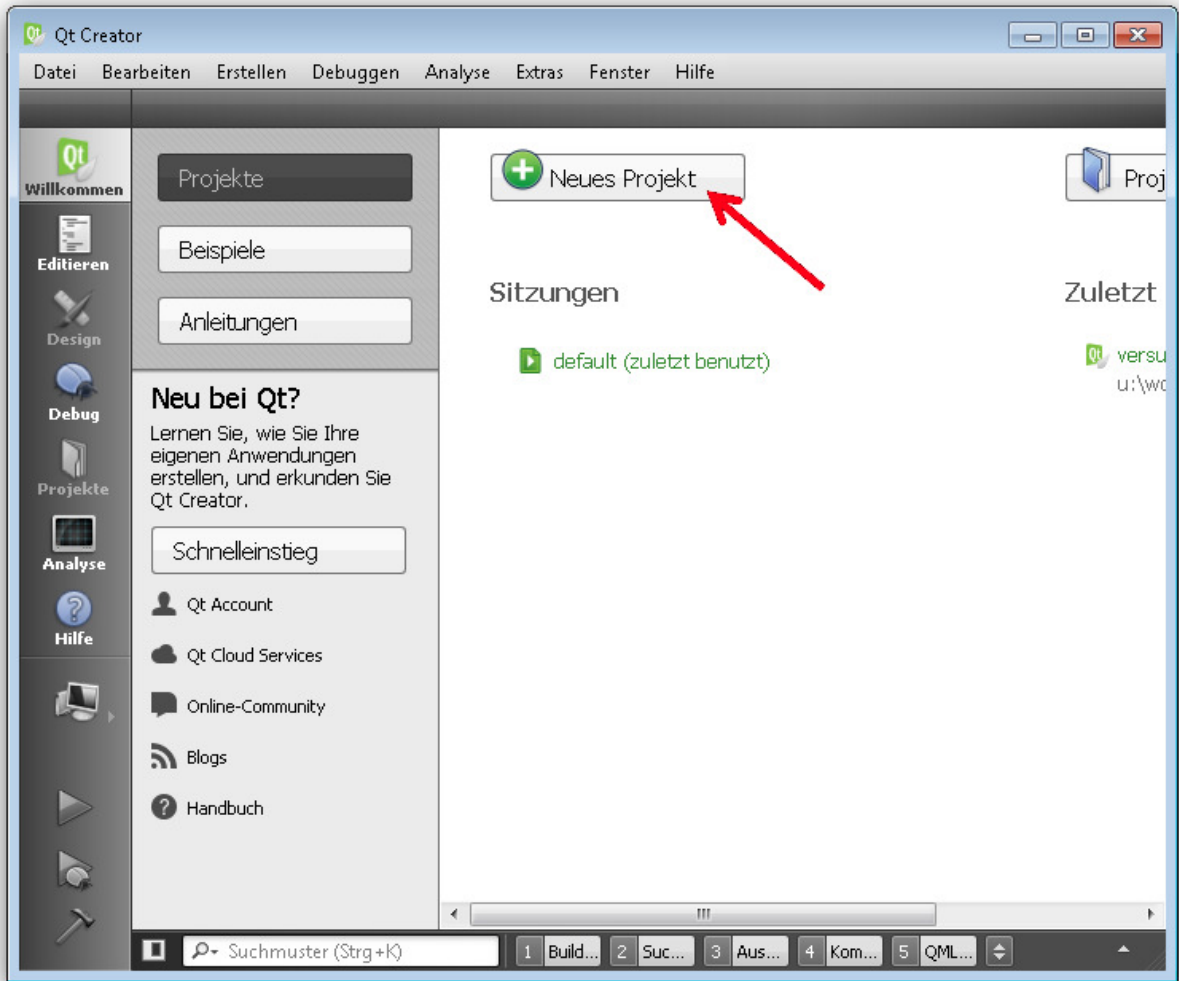
### Ziel

Es soll ein einfaches C-Programm mithilfe der IDE fehler- und warnungsfrei erstellt und ausgeführt werden.

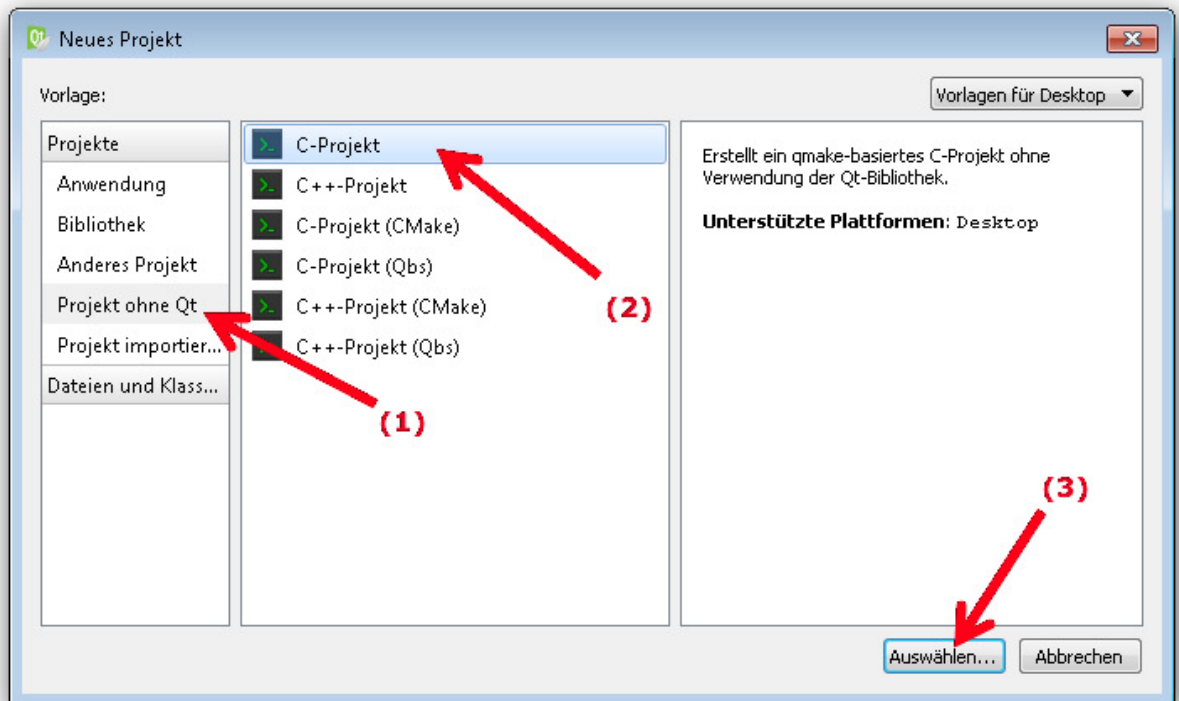
### Durchführung

- Starten Sie das Programm **QTcreator (Community)**.

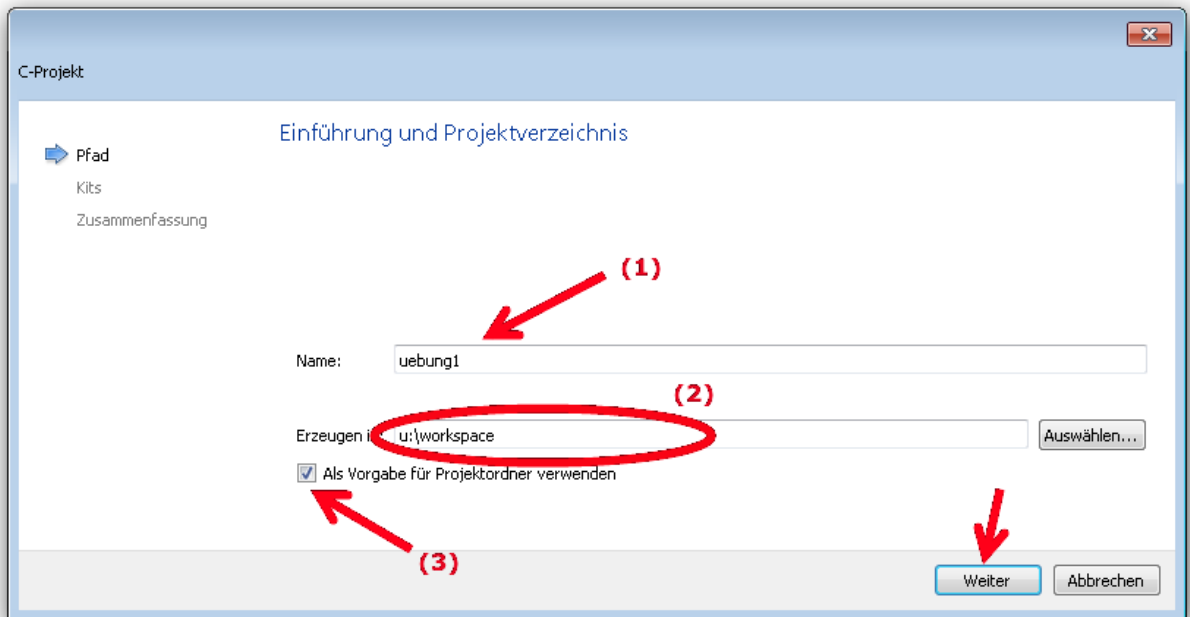
- Erstellen Sie ein neues Projekt (bzw. Arbeitsbereich):



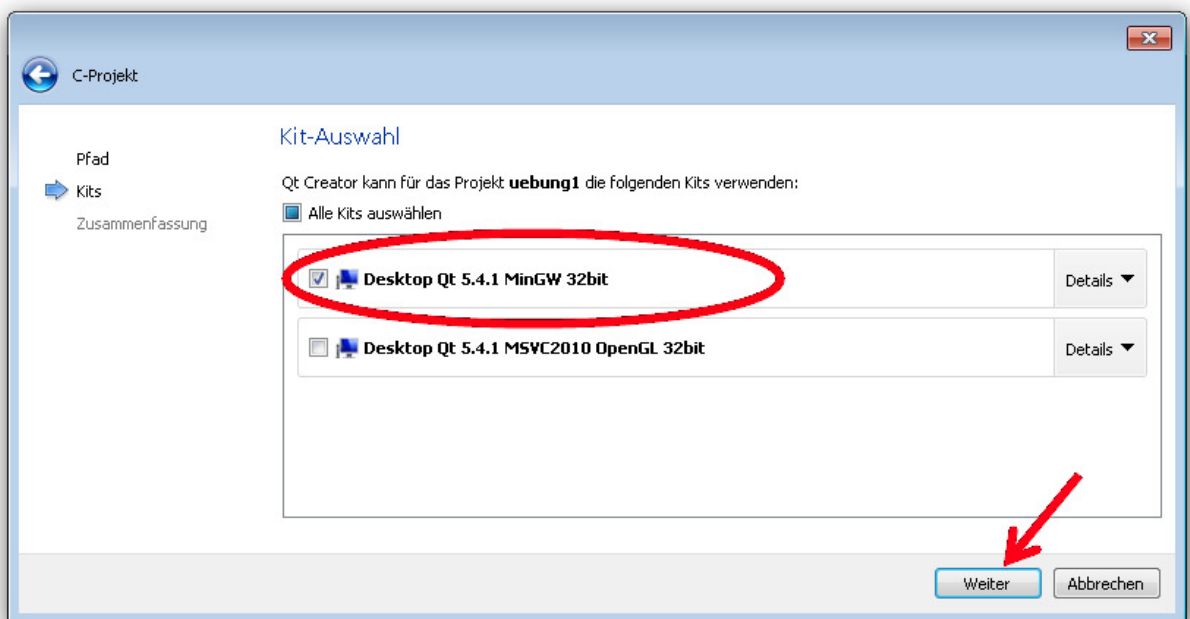
- Wählen Sie nun den Projekttyp (Projekt ohne QT → C-Projekt):



- Wählen Sie nun als Projektnamen **uebung1** und legen als Arbeitsbereichverzeichnis **U:\workspace** fest.  
Für die zukünftigen Projekte ist es sinnvoll das Projektverzeichnis für zukünftige Projekte als Standardverzeichnis zu definieren **(3)**.

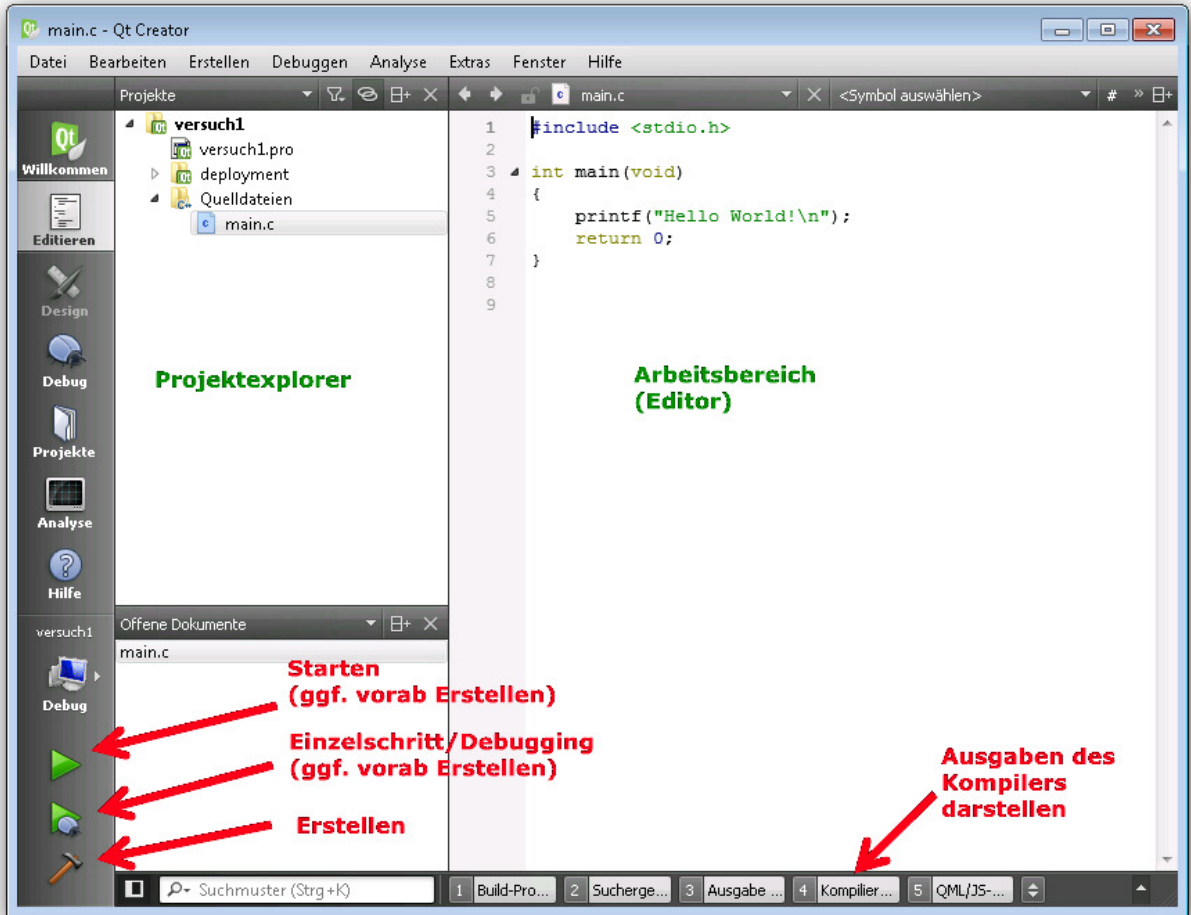


- Als nächstes wird das **Compiler-Toolkit** (d. h. der Werkzeugkasten zum Erstellen von ausführbaren Dateien) gewählt.  
Stellen Sie sicher, dass das **MinGW-Toolkit** selektiert wurde. (Auf die Auswahl des **Microsoft-Visual-C-Toolkit** kann verzichtet werden.)



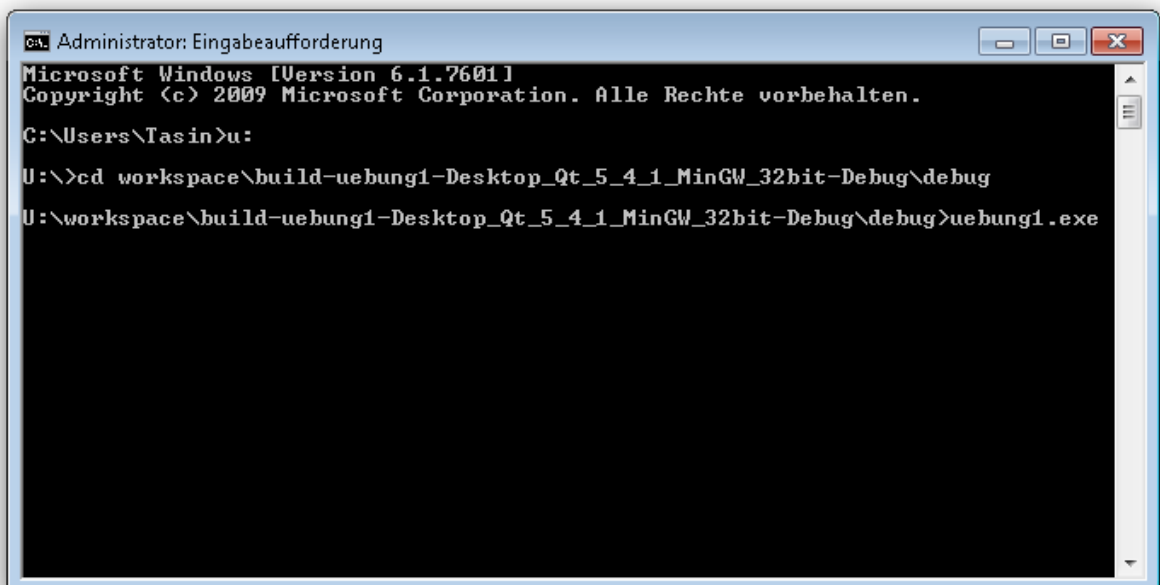
- Die letzte Dialogbox kann ohne Änderung abgeschlossen werden.  
Klicken Sie dazu auf den Button **Abschließen**.

- **Erstellen und Starten** Sie das Beispielprogramm:

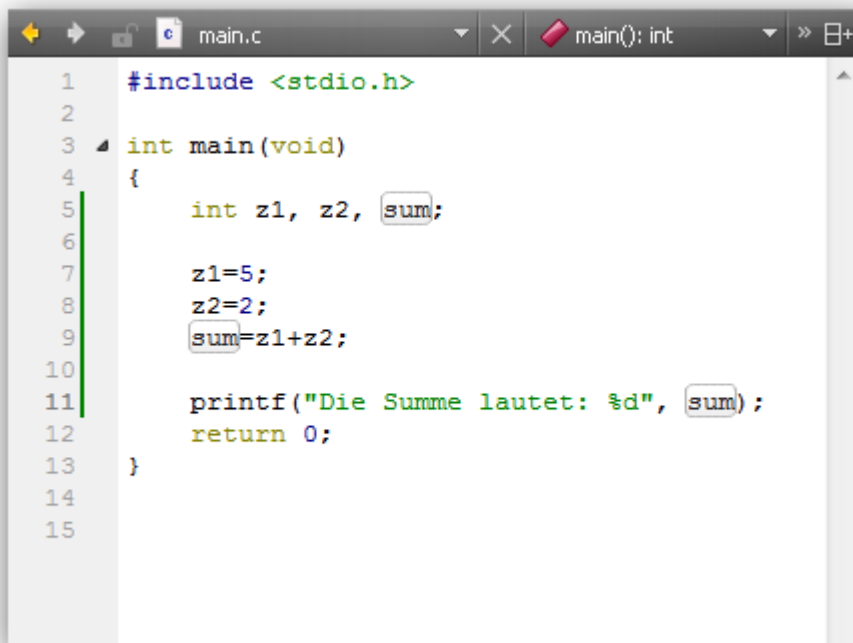


**Frage B1:** Was können Sie im **Ausgabebereich** durch Klicken auf **4 Kompilierung** feststellen?

- Öffnen Sie nun eine **Eingabeaufforderung** ("DOS-Box"/"Command Prompt")  
*Start Knopf* → *Alle Programme* → *Zubehör* auf *Eingabeaufforderung* klicken  
(*Start Button* → *All Programs* → *Accessories* auf *Command Prompt* klicken).
- Wechseln Sie nun in der Eingabeaufforderung das Laufwerk durch Eingabe von **U:** und quittieren diese mit **<Enter>**. Tätigen Sie die Eingaben, wie Sie im Screenshot dargestellt werden:



- Ändern Sie nun den Quellcode wie folgt ab:



```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int z1, z2, sum;
6
7     z1=5;
8     z2=2;
9     sum=z1+z2;
10
11     printf("Die Summe lautet: %d", sum);
12     return 0;
13 }
14
15
```

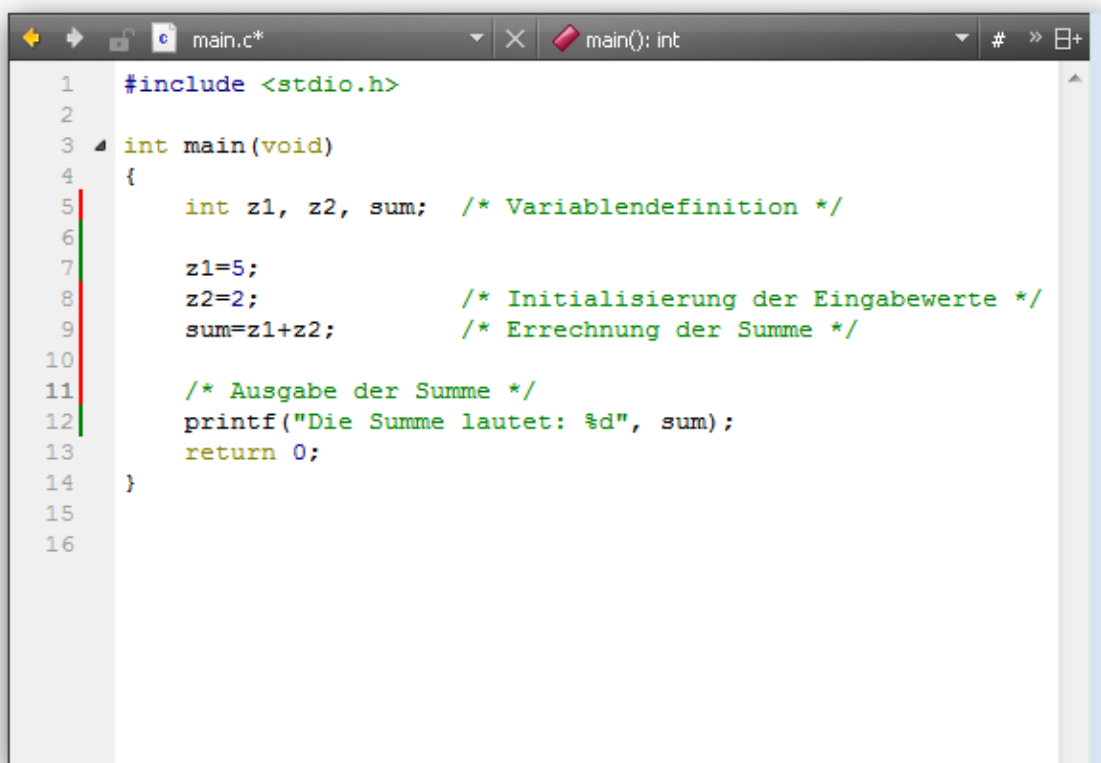
Speichern Sie die Datei nun ab.

**Frage B2:** Woran kann man erkennen, dass die Datei noch nicht auf der Festplatte gesichert wurde?

- Wiederholen Sie den Vorgang des Erstellens, Starten der ausführbaren Datei über **QT Creator** und über die Eingabeaufforderung.

Korrigieren Sie evtl. Warnungen und Fehler indem Sie die Meldungen im Ausgabebereich interpretieren und korrigieren.

- Ergänzen Sie Ihren Quellcode um einige Kommentare und wiederholen Sie Erstellen und Starten des Programms:

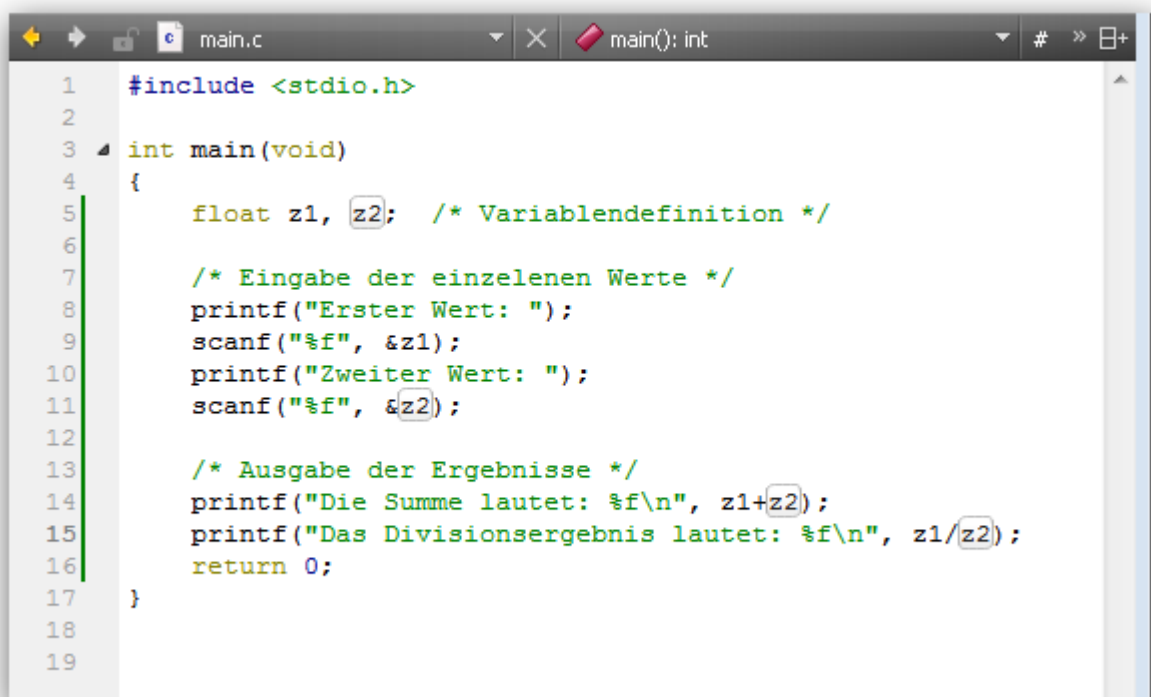


```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int z1, z2, sum; /* Variablendefinition */
6
7     z1=5;
8     z2=2;           /* Initialisierung der Eingabewerte */
9     sum=z1+z2;     /* Errechnung der Summe */
10
11     /* Ausgabe der Summe */
12     printf("Die Summe lautet: %d", sum);
13     return 0;
14 }
15
16
```

- Fügen Sie nun eine neue Variable **div** der Variablendefinition hinzu. Sie soll ebenfalls vom Typ **int** sein, das Ergebnis der Division von **z1** mit **z2** beinhalten und danach den Variablenwert zusätzlich zur Summe ausgeben.  
Ändern Sie den Quellcode entsprechend ab und starten Sie Ihr Programm mit unterschiedlichen Eingabewerten.

**Frage B3:** Was können Sie über das Ergebnis sagen?

- Ändern Sie nun den Datentyp in der **Variablendefinition** von **int** zu **float** ab.
- Teilen Sie den **printf**-Funktionsaufrufen mit, dass nun eine Fließkommazahlenausgabe ausgeführt werden soll indem Sie die Formatspezifizierer **%d** in **%f** ändern.
- Erstellen Sie den neuen Quellcode und korrigieren Sie eventuelle Fehler/Warnungen. Starten Sie Ihr Programm nach erfolgreichem Erstellen.
- Ändern Sie den Quellcode erneut ab:



```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      float z1, z2; /* Variablendefinition */
6
7      /* Eingabe der einzelnen Werte */
8      printf("Erster Wert: ");
9      scanf("%f", &z1);
10     printf("Zweiter Wert: ");
11     scanf("%f", &z2);
12
13     /* Ausgabe der Ergebnisse */
14     printf("Die Summe lautet: %f\n", z1+z2);
15     printf("Das Divisionsergebnis lautet: %f\n", z1/z2);
16     return 0;
17 }
18
19
```

- Sichern Sie den neuen Quellcode, Erstellen das Programm und lassen es unter der **Eingabeaufforderung** laufen.  
**Frage B4:** Durch welche Funktionsaufrufe werden die Variablen mit Daten versorgt?  
**Frage B5:** Was passiert wenn diese Aufrufe in Kommentare umgewandelt werden und welche Ausgabe erhält man dann?