

2. Übung Informatik PG 1

In dieser Übung starten wir mit der C-Programmierung. Das Ziel dieser Übung ist es, erste Erfahrungen mit der Programmierungsumgebung sowie der C-Syntax zu sammeln. Programmieren ist immer auch ausprobieren wie Code funktioniert, daher werden wir in dieser Übung viel vermeintlich Sinnloses ausprobieren. Führen Sie bei Ihren Versuchen auch immer absichtlich Fehler herbei, um ein Gefühl dafür zu entwickeln, wie Fehlermeldungen aussehen. Hinweis: Der Begriff Deklaration schließt im Folgenden die Definition von Variablen mit ein.

1. Hello, World!

- Erstellen Sie eine leere Quellcode-Datei, schreiben Sie den Hello-World-Code aus der Vorlesung ab und kompilieren Sie diesen. Rufen Sie das Programm auf. Verwenden Sie hierfür zunächst einen Texteditor und die Konsole sowie anschließend eine IDE.
- In der Vorlesung wurde angesprochen, dass das Steuerzeichen `\n` für einen Zeilenvorschub steht. Googlen Sie weitere Steuerzeichen und probieren Sie deren Funktion aus.
- Deklarieren und initialisieren Sie Variablen verschiedener Datentypen und geben Sie diese über die Printfunktion aus.
- Was passiert bzw. welche Fehlermeldungen erhalten Sie, wenn Sie Syntaxteile weglassen (bspw. die Headerdatei, das Semikolon), wenn Sie einzelne Anführungszeichen verwenden statt doppelte, ... was Ihnen so einfällt.

2. Debugging

Eine Hauptaufgabe von Programmierer:innen ist das Debuggen von Code, mittels Debugger sowie `printf`-Statements.

Schreiben Sie Code der die verschiedenen Datentypen im Verlauf des Programms verändert und testen Sie das Debuggingtool Ihrer IDE. Verwenden Sie außerdem `printf`-Statements um die Veränderung der Variablen nachzuvollziehen. Ziel dieser Übung ist es, dass Sie erste Debuggingsschritte gehen und eine Möglichkeit kennenlernen nachzuvollziehen, was im Code passiert. Dies ist jetzt noch recht trivial, wird aber mit zunehmender Code-Komplexität interessanter und wichtiger.

3. Variablen und Datentypen

- Was passiert, wenn Sie eine Variable nur deklarieren aber nicht initialisieren und dann per `printf`-Funktion ausgeben (oder Sie sich den Wert mittels Debugger anschauen)?
- Was passiert, wenn Sie eine Variable eines bestimmten Datentyps falsch initialisieren bzw. außerhalb des Wertebereichs? Nutzen Sie hierfür die `printf`-Funktion sowie den Debugger.
- In der Vorlesung sind wir kurz auf den Character-Datentyp und die ASCII-Tabelle eingegangen. Probieren Sie die Typumwandlung selbst aus, indem Sie die in der ASCII-Tabelle hinterlegten Zeichen sowohl als Zeichen als auch als Zahl verwenden.
- Was passiert, wenn Sie zwei Zeichen (`char`) addieren und als Integer ausgeben lassen?

- Testen Sie die implizite und explizite Typumwandlung von Zahlen. Welches Ergebnis erhalten Sie bei den folgenden Berechnungen? (Nutzen Sie printf-Statements, um sich die Ergebnisse anzeigen zu lassen).

```
int main() {  
    int x = 5, y = 2;  
    float result = 0.0;  
    result = (float) x / y;  
    result = float (x / y);  
}
```

1. Scanf-Funktion

- Testen Sie die scanf-Funktion mit verschiedenen Datentypen. Untersuchen Sie den Unterschied wenn Sie Zahlen einlesen im Gegensatz zu Zeichen (char), insbesondere bei mehreren Abfragen nacheinander. Nutzen Sie die in der Vorlesung vorgestellte Methode um den Speicher nach dem Einlesen von Zeichen zu leeren.
- Welchen Wert erhalten Sie, wenn Sie kein Zeichen mittels scanf einlesen, sondern die Entertaste verwenden und sich den eingelesenen Wert als Integer ausgeben lassen? (Denken Sie an die ASCII-Tabelle).
- Was passiert, wenn Sie Umlaute eingeben?