

# Praktikumstermin Nr. 1, INF: Visual Studio, HalloWelt Programm

## Ziel dieses Praktikumstermins

Sie machen sich mit der Entwicklungsumgebung Visual Studio vertraut. Ferner analysieren Sie einige Grundregeln der Programmiersprache C++.

**Sie finden in Ilias neben dem Aufgabendokument (dieses Dokument hier) noch ein weiteres Dokument mit zusätzlichen Informationen zur Durchführung dieses Praktikumsversuchs.**

Bitte beachten Sie die empfohlenen Einstellungen für Visual Studio, die sich in diesem begleitenden Material finden. Diese Einstellungen müssen Sie zum größten Teil manuell konfigurieren. Am besten im Rahmen dieses Praktikumsversuchs.

*Ziel hier: Vertraut machen mit der benutzten Entwicklungsumgebung, Analysieren einiger Grundregeln von C++.*

## Abgabemethodik bei diesem Praktikumstermin

Nachdem Sie alle Aufgaben für sich alleine erfolgreich durchgearbeitet haben, gehen Sie die Aufgaben noch einmal gemeinsam mit der Praktikumsbetreuung durch und beschreiben, was sie beim Durcharbeiten der Aufgaben für Ergebnisse beobachtet haben.

Ob Sie dabei die notwendigen Änderungen im Programm und das entsprechende Neucompilieren und Neustarten des Programms noch einmal wiederholen oder ob Sie das Programmverhalten aus ihrem Gedächtnis bzw. basierend auf etwaigen Notizen, die Sie vielleicht gemacht haben, erzählen entscheiden Sie nach Bedarf gemeinsam mit der Praktikumsaufsicht.

# Aufgabe INF-01.01: Projektmappe und Projekt anlegen

Starten Sie Visual Studio. Unter Windows 10: Klick auf das Windows Icon links unten, dann „blindes“ Eintippen des Texts **visual**, gefolgt vom Anklicken des **Visual Studio** Eintrags.

Legen Sie in Visual Studio ein neues Projekt **HalloWelt** (eingeben unter **Projektname**) und eine neue Projektmappe **GIP-2021-2022** (eingeben unter **Projektmappenname**) an.

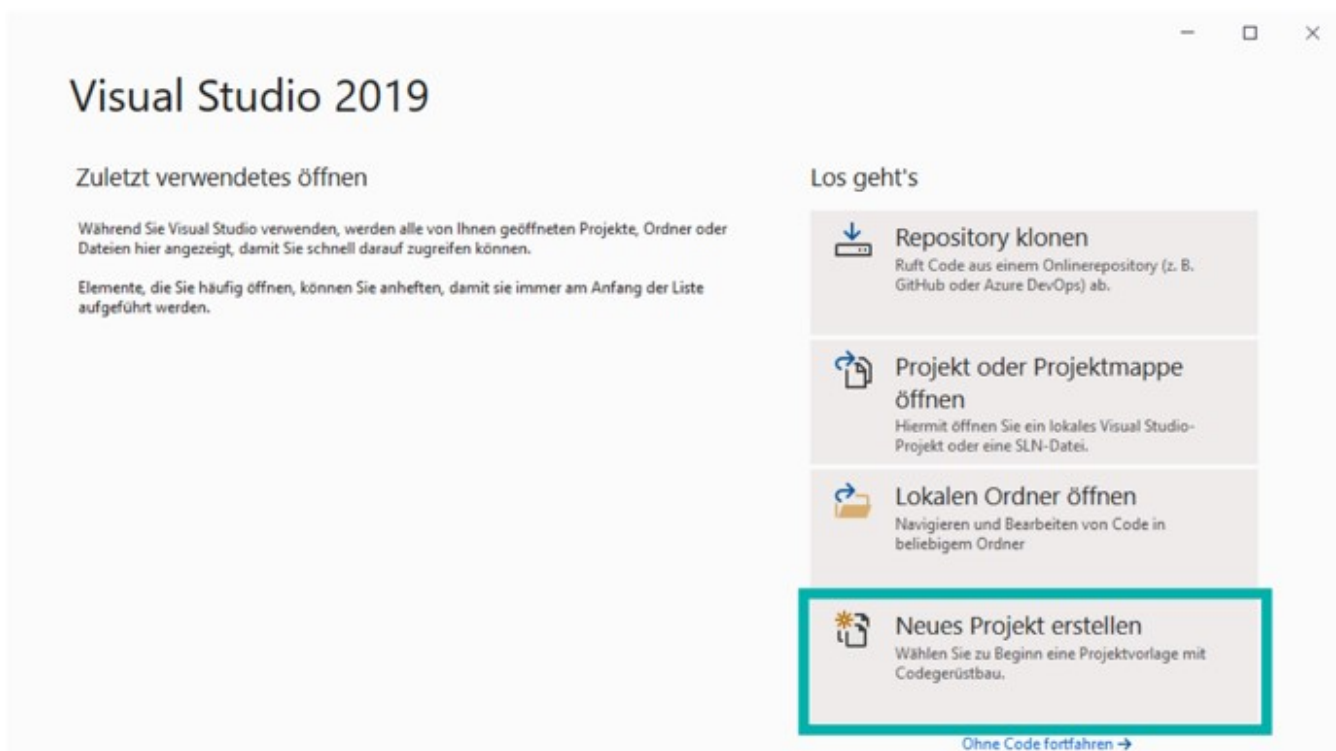


Abbildung 1. Neues Projekt erstellen

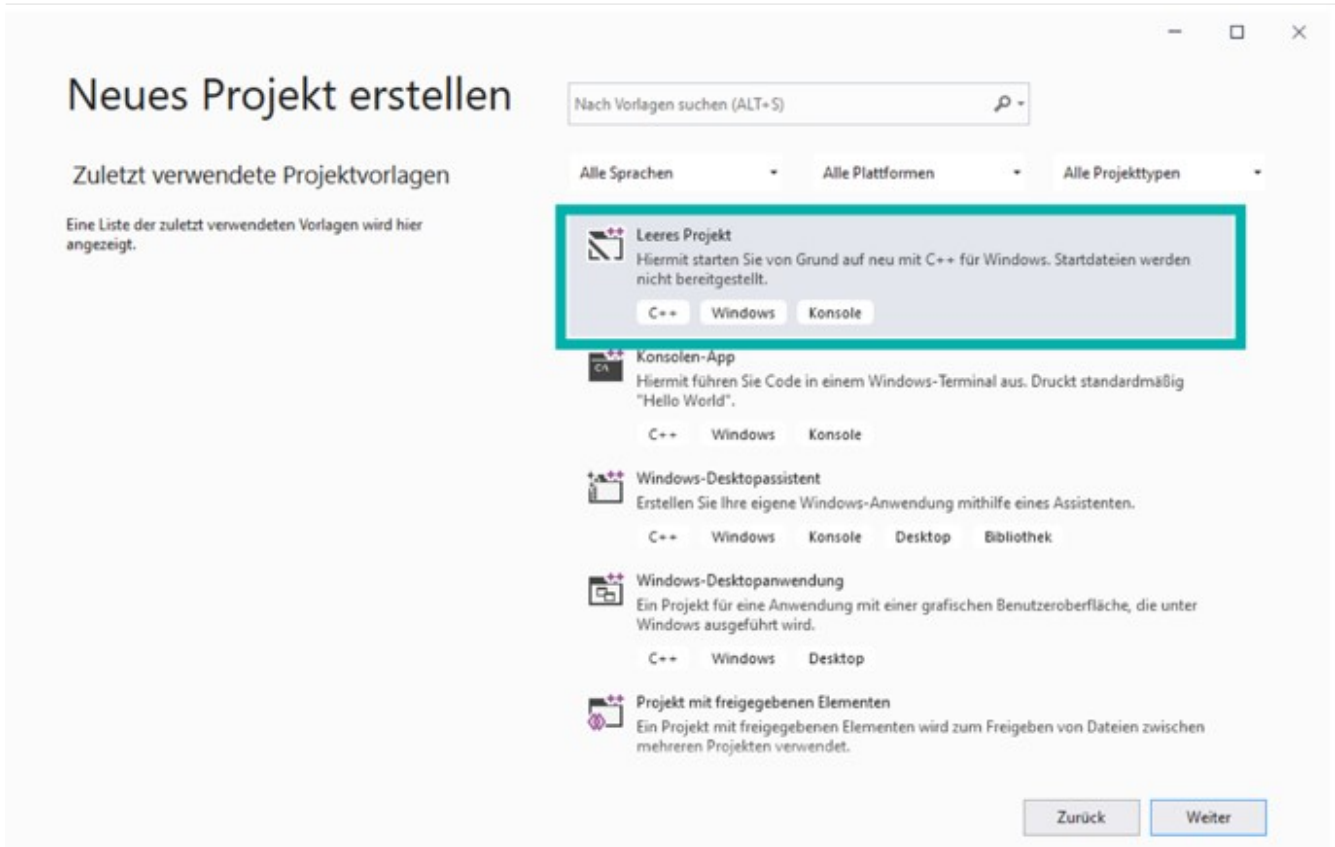


Abbildung 2. Leeres Projekt auswählen

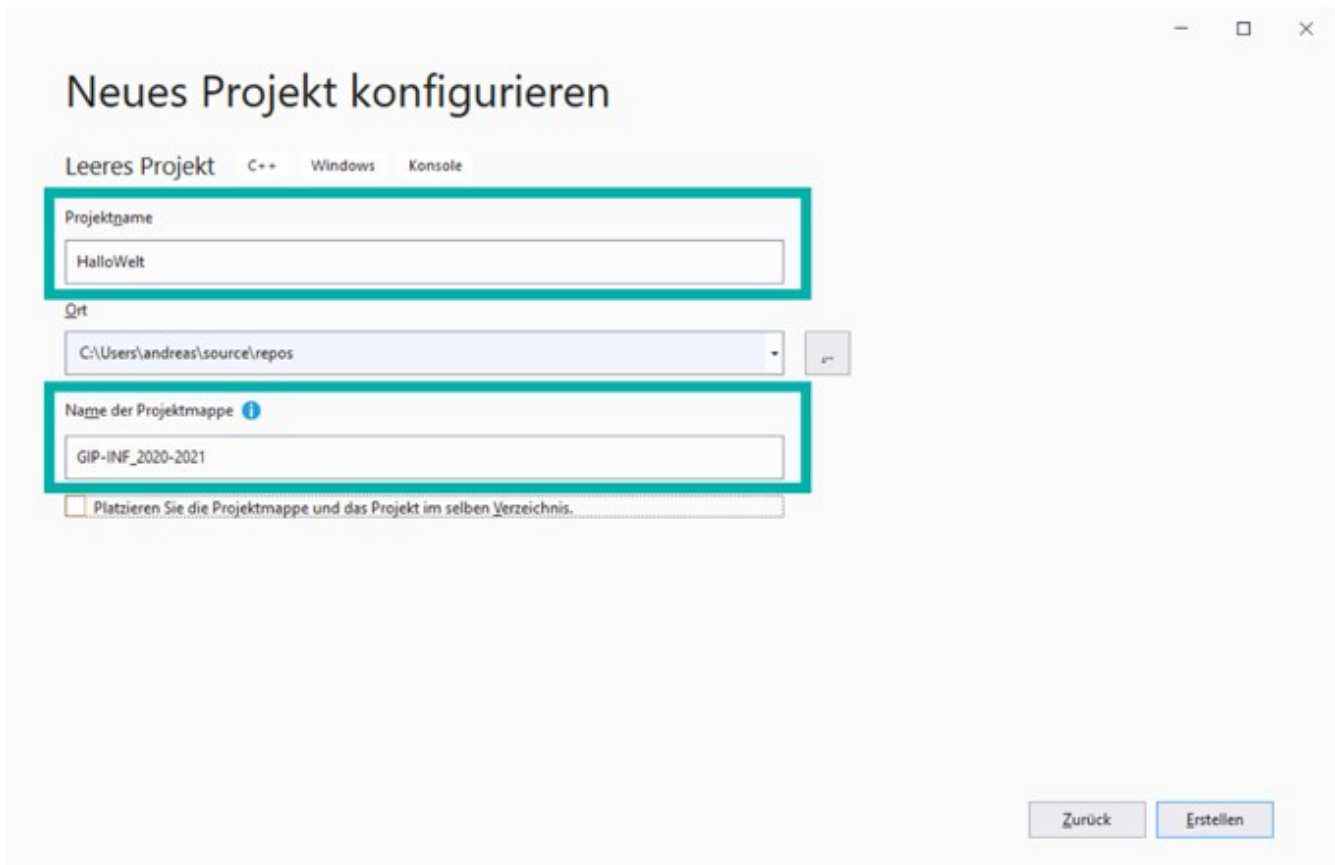


Abbildung 3. Neues Projekt konfigurieren

# Aufgabe INF-01.02: HalloWelt Programm

Geben Sie das C++ HalloWelt-Programm ein.

Quelltext 1. HalloWelt C++-Programm

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hallo Welt!" << endl;

    system("PAUSE");
    return 0;
}
```

Fügen Sie vor der Zeile `return 0;` die Zeile `system("PAUSE");` in ihr Programm ein.

Compilieren und starten Sie das Programm aus Visual Studio heraus.

Testlauf

```
Hallo Welt!
Drücken Sie eine beliebige Taste . . .
```

## Aufgabe INF-01.03: Relevanz `return 0` & `system("PAUSE")`

Setzen Sie die Zeilen `return 0;` und `system("PAUSE");` in einen Blockkommentar. Was ist der Unterschied in der Ausführung verglichen mit Aufgabe INF-01.02 und welchen Rückgabewert des Programms beobachten Sie in Visual Studio?

## Aufgabe INF-01.04: Blockkommentare; Codeblöcke auskommentieren

Ändern Sie den Blockkommentar aus Aufgabe INF-01.03 so, dass alle Zeilen im Rumpf des Hauptprogramms in den Blockkommentar eingeschlossen sind. Führen Sie das Programm aus. Was beobachten Sie? Was können wir also daraus folgern für ein C++ Hauptprogramm mit einem leeren Rumpf? Compiliert so ein Programm? Und wie sieht



Dieses Vorgehen, einen Teil des C++ Codes über Kommentare zu „deaktivieren“, nennt man das Auskommentieren dieses (Teils des) Codes.

## Aufgabe INF-01.05: Zeilennummerierung

Schalten Sie im Visual Studio Editor die automatische Zeilennummerierung an: Menü **Extras** ⇒ **Optionen** ⇒ **Text-Editor** (in der baumartigen Liste ganz links) ⇒ **C/C++** ⇒ **Allgemein**, dort dann ein Häkchen setzen bei **Zeilennummern** unter **Einstellungen**.



In der Baumdarstellung müssen Sie gegebenenfalls Knoten „aufklappen“ durch Linksklick auf das kleine Dreieck.



Die Zeilennummerierung sollte ab nun in den Praktika immer eingeschaltet werden. Dies erleichtert die Diskussion über ein Programm.

## Aufgabe INF-01.06: Relevanz von Zeilenumbrüchen

In C++ werden Anweisungen mit einem Semikolon abgeschlossen. Verändern Sie den Quellcode so, dass Sie die C++ Anweisungen über mehrere Zeilen „strecken“, indem Sie z.B. bei **return 0;** das **return**, die **0** und das Semikolon in aufeinanderfolgende Zeilen schreiben. Compiliert und läuft das Programm noch?

## Aufgabe INF-01.07: Semikolon terminiert Anweisungen

Nehmen Sie nun das Semikolon hinter einer der Anweisungen ganz weg. Compiliert und läuft das Programm noch?

Beobachten Sie die Syntaxprüfung, die schon in den Quelltext-Editor eingebaut ist: Suchen Sie rot unterschlingelte Teile des Quelltexts, fahren Sie mit der Maus darüber und warten Sie zwei Sekunden (Mauszeiger nicht bewegen). Was meldet das kleine Pop-Up für die Fehlerstelle?

## Aufgabe INF-01.08: Mehrere Anweisungen in einer Zeile

Ergänzen Sie das Semikolon wieder und schreiben Sie mehrere Anweisungen in eine Textzeile des Quelltexts. Compiliert und läuft das Programm noch?

## Aufgabe INF-01.09: Anweisungsblöcke

Setzen Sie jeden C++ Befehl (der sich ggf. über mehrere Zeilen des Quelltexts erstreckt) in eigene geschweifte Klammern `{...}`. Compiliert das Programm? Ändert sich etwas an der Programmausführung?

## Aufgabe INF-01.10: Einrückungen

Einrückungen des C++ Programmcodes (z.B. im Rumpf des Hauptprogramms) dienen nur dazu, die Verständlichkeit des Quelltexts für den Programmierer zu erhöhen. Verifizieren Sie dies, indem Sie zuerst verschiedene Zeilen des Quelltexts verschieden stark einrücken und anschließend alle Einrückungen innerhalb des Quelltexts ganz wegnehmen.

Setzen Sie dann die Einrückungen zurück, indem Sie über das Menü **Bearbeiten** ⇒ **Erweitert** ⇒ **Dokument formatieren** eine einheitliche Einrückung der Zeilen erzielen.

## Empfohlene Visual Studio Einstellungen

Das Setzen dieser Einstellungen ist nicht Teil des Praktikums. D.h. Sie brauchen nicht „vorzuzeigen“, dass Sie diese Einstellungen geändert haben. Diese Einstellungen werden aber in kommenden Praktikumsversuchen benötigt bzw. erleichtern Ihnen die Entwicklung Ihrer Programme und uns die Betreuung Ihres Praktikums.

Welche Einstellungen wie geändert werden sollen finden Sie in den begleitenden Unterlagen zum heutigen Praktikumsversuch.