

Allgemeine Hinweise

Parallel zum Präsenzunterricht erhalten Sie wöchentlich ein Arbeitsblatt wie dieses hier. Darin finden Sie jeweils einige Aufgaben zu den aktuell behandelten Themen. Die Aufgaben dienen zum einen der Festigung des gelernten Stoffs, aber zum anderen auch zu dessen Vertiefung. Alle Aufgabenblätter sind integraler Bestandteil der Lehrveranstaltung und deshalb auch prüfungsrelevant. Es lohnt sich also auf jeden Fall, die Aufgaben regelmässig zu bearbeiten.

Die Aufgaben kommen in zwei unterschiedlichen Varianten und können anhand eines Symbols, mit dem sie gekennzeichnet sind, unterschieden werden:

- ✍ Durch Lösen dieser *Kontrollaufgaben* wird das erworbene Wissen gefestigt und Verständnislücken können aufgedeckt werden. Versuchen Sie jeweils, diese Aufgaben zuerst selbstständig auf Papier zu lösen und vergleichen Sie dann Ihre Antworten mit der Musterlösung.
- 🔍 Bei diesen *Lernaufgaben* geht es darum, eine in der Vorlesung behandelte Thematik weiter zu vertiefen. Typischerweise verlangt die Aufgabe nach einer kleinen kreativen Idee, um das gestellte Problem zu lösen. Falls Sie nicht drauf kommen, tauschen Sie sich mit Ihren Kommilitonen aus.
- ✓ Ausführliche *Musterlösungen* zu den gestellten Aufgaben werden zeitgleich mit der Veröffentlichung des entsprechenden Arbeitsblatts in einem separaten Dokument zur Verfügung gestellt. Auch wenn Sie alle Aufgaben lösen konnten, lohnt es sich trotzdem, einmal reinzuschauen.

Mit der folgenden ersten Lernaufgabe wollen wir sicherstellen, dass Sie C++-Projekte insbesondere auch mit mehreren Quellcode-Dateien kompilieren können. Da sich das genaue Vorgehen je nach gewählter Entwicklungsumgebung stark unterscheidet, wird es hierfür ausnahmsweise keine Lösung geben.

🔍 **Aufgabe 0.1** Erstellen Sie in Ihrer Entwicklungsumgebung folgende drei Dateien:

```
main.cpp

#include <iostream>
#include "other.h"

// Hallo-Funktion aus main.cpp
void hello1() {
    std::cout << "Main: Hello!\n";
}

// Main-Funktion
int main() {
    hello1();
    hello2();
}
```

```
other.h

// Deklaration der Hallo-Funktion
// aus der Datei other.cpp
void hello2();
```

```
other.cpp

#include <iostream>
#include "other.h"

// Hallo-Funktion aus other.cpp
void hello2() {
    std::cout << "Other: Hello!\n";
}
```

Kompilieren Sie Ihr Projekt mit den drei vorgegebenen Dateien oben und führen Sie dann das generierte Programm aus. Wie man diese zwei Schritte genau ausführt, hängt von der von Ihnen gewählten Entwicklungsumgebung ab.

Falls Sie zwei verschiedene Begrüssungen angezeigt bekommen, haben Sie alles richtig gemacht. Auf analoge Weise können Sie nun C++-Projekte mit beliebig vielen Dateien erstellen. Das heisst, für jede `cpp`-Datei gibt es typischerweise eine entsprechende `h`-Datei, in der die definierten Funktionen (und später auch Klassen und weiteres) deklariert werden, so dass durch Inkludieren der `h`-Datei diese Funktionen auch in anderen Dateien des Projekts bekannt und benutzbar gemacht werden können.