

# Algorithmen und Datenstrukturen I

## Blatt 1

Prof. Dr. Oliver Braun

Fakultät für Informatik und Mathematik  
Hochschule München

Letzte Änderung: 09.12.2019 16:28

### Aufgabe 1 — GitHub et al.

Für die Bearbeitung der Praktikumsblätter verwenden wir GitHub. Sie benötigen einen beliebigen GitHub-Account der keinen Zusammenhang mit Ihrer Hochschulken- nung haben muss. Damit ich Ihnen aber am Ende einen Schein geben kann, muss ich natürlich wissen wer sich hinter welchem Pseudonym verbirgt. Dazu gehen Sie auf <https://classroom.github.com/g/8jmjdXo-> und wählen Ihre E-Mail-Adresse aus. Sollte Ihre E-Mail-Adresse nicht aufgeführt sein, waren Sie zum Zeitpunkt der Erstellung der Liste nicht im ZPA für diese Veranstaltung eingeteilt. Daher können Sie auch keinen Schein von mir bekommen. Sollte das ein Fehler sein, nehmen Sie bitte umgehend mit mir Verbindung auf.

Treten Sie anschließend dem Team `ws19` bei. Dadurch erhalten Sie Zugriff auf unser gemeinsames Repository <https://github.com/ob-algdati-ws19/ob-algdati-ws19>.

Darüber hinaus verwenden wir einen Jenkins unter <https://terraform.cs.hm.edu/>. Auf diesem müssen Sie sich auch mit Ihren GitHub-Credentials einloggen. Die Repositories auf GitHub werden `private` sein. Auf dem Jenkins sehen Sie jedoch alle Builds.

### Aufgabe 2 — C++-IDE

Machen Sie sich einer Entwicklungsumgebung für C++ vertraut. In den Laboren ist unter anderem CLion auf der [DevBox](#) installiert. Sie können natürlich auch eine beliebige andere IDE nutzen. Voraussetzung für die spätere Abgabe ist allerdings, dass sich die Lösungen mit CMake erstellen lassen. CLion hat CMake gleich eingebaut. Mit einem

[JetBrains-Account für Studierende](#) können Sie CLion kostenlos herunterladen und nutzen. Voraussetzung ist die Verwendung Ihrer @hm.edu-E-Mail-Adresse für den Account. Weiterhin ist [VS Code](#) mit dem [C++-Plugin](#) zu empfehlen. Dies werde ich in der Vorlesung verwenden.

Versuchen Sie zunächst das “Hello World”-Programm zum Laufen zu bringen:

```
#include <iostream>

int main() {
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

**Anmerkung für CLion auf eigenem Windows-Rechner:** Das in CLion gebundelte CMake macht u.U. Probleme. Infos wie Sie den CLion unter Windows vernünftig zum Laufen bekommen, finden Sie [hier](#).

## Schöner Code

Programmieren, also Schreiben von Quelltext, ist eine kreative Aufgabe. Für uns Programmierer und Programmiererinnen ist der Quelltext bedeutender als das damit erzeugte Programm. Daher sollte der Code auch immer gut zu lesen sein. Ein wichtiges Kriterium dabei ist die einheitliche (einheitlich mindestens innerhalb eines Projektes) Formatierung. Den Code selbst zu formatieren, ist allerdings nicht mehr wirklich zeitgemäß und natürlich auch fehleranfällig. Konfigurieren Sie daher Ihre IDE so, dass sie bei jedem Speichern den Code formatiert. Verwenden Sie eine **übliche** Formatierung und erfinden Sie keine neue. Ich lasse beispielsweise meinen C++-Code gemäß dem [Google-Style](#) formatieren.

## Aufgabe 3 — Fehler finden

Das Repository für diese Aufgabe bekommen Sie über die URL <https://classroom.github.com/a/TyYfvJrL>.

Versuchen Sie das im Repository enthaltene C++-Programm zum Laufen zu bekommen, indem Sie die Fehler finden. Es soll die Fakultäten aller Zahlen von 1 bis 20 berechnen und ausgeben.

Bei vielen Fehlern wird Ihnen der Compiler helfen. Versuchen Sie insbesondere zu verstehen wie das Programm funktioniert und was das alles bedeutet. Fragen Sie im Zweifelsfall ihre Kommilitonen oder mich.

## Aufgabe 4 — Rekursion

Ersetzen Sie im Programm aus Aufgabe 4 die iterative Berechnung der Fakultät durch eine rekursive.