

0. ORGANISATION

Das Modul Informatik 1

Das Modul „Informatik 1“ besteht aus **zwei** Lehrveranstaltungen

Konzepte der Informatik (Kdl) vermittelt die theoretischen Grundlagen von Algorithmen und Datenstrukturen

Programmierkurs 1 (PK1) setzt Algorithmen und Datenstrukturen in der Programmiersprache C++ praktisch um

Achtung!

- ▶ Beide Lehrveranstaltungen bilden eine Einheit
- ▶ PK1 greift Stoff aus Kdl auf und wendet diesen praktisch an
- ▶ Es empfiehlt sich nicht, PK1 unabhängig von Kdl zu besuchen!

Literatur

Dieser Programmierkurs stellt die meisten Inhalte **exemplarisch** vor

- ▶ **Folien**
- ▶ **Code-Beispiele**

Für die Details gibt es genügend andere Referenzen und Quellen

- ▶ **Internet**
 - <https://en.cppreference.com>
 - Google und Co.
- ▶ **Bücher**
 - z.B. **Bjarne Stroustrup**: *Eine Tour durch C++*. 2023, mitp Verlags GmbH. ISBN 978-3-7475-0625-7.



Koordinaten

► Material

Ilias Folien und Forum

GitLab Übungsaufgaben

► Termine

- Programmierkurs, Dienstag, 13:30–15:00, R 711
- Tutorien
 - Donnerstag, 11:45–13:15, M 628
 - Donnerstag, 11:45–13:15, F 429
 - Donnerstag, 13:30–15:00, M 630
 - Donnerstag, 13:30–15:00, ML 630
 - Donnerstag, 17:00–18:30, ML 630
 - Freitag, 08:15–09:45, P 602
- Übungsteilnahme ist nicht obligatorisch
- Flexible Übungswahl (Anmeldung in ZEuS zur Platzabschätzung)

Personen

- ▶ Sabine Storandt
- ▶ Michael Grossniklaus
- ▶ Ruoying Li
- ▶ Gregor Diatzko
- ▶ Jasper Krauter

Übungen

- ▶ Übungsaufgaben werden in **Git** bzw. **GitLab** gestellt und auch dort abgegeben und bewertet (cf. Schlüsselqualifikation, 1. Semester)
- ▶ 10+1 Übungsblätter

Übungen

- ▶ Übungsaufgaben werden in **Git** bzw. **GitLab** gestellt und auch dort abgegeben und bewertet (cf. Schlüsselqualifikation, 1. Semester)
- ▶ 10+1 Übungsblätter

Reguläre Übungsblätter (1-10)

- ▶ Jeweils drei Aufgaben (10 P. + 5 P. + 5 P.)
- ▶ Automatische Bepunktung (direkt nach Upload)
- ▶ Kriterien: Korrektheit, Code Style, Lines of Code
- ▶ **Einzelabgabe!**



Übungen

- ▶ Übungsaufgaben werden in **Git** bzw. **GitLab** gestellt und auch dort abgegeben und bewertet (cf. Schlüsselqualifikation, 1. Semester)
- ▶ 10+1 Übungsblätter

Reguläre Übungsblätter (1-10)

- ▶ Jeweils drei Aufgaben (10 P. + 5 P. + 5 P.)
- ▶ Automatische Bepunktung (direkt nach Upload)
- ▶ Kriterien: Korrektheit, Code Style, Lines of Code
- ▶ **Einzelabgabe!**



Zum Bestehen müssen pro Woche ≥ 8 von 20 Punkten (2 Strikes) und insgesamt ≥ 140 von 200 Punkten erzielt werden.

Übungen

- ▶ Übungsaufgaben werden in **Git** bzw. **GitLab** gestellt und auch dort abgegeben und bewertet (cf. Schlüsselqualifikation, 1. Semester)
- ▶ 10+1 Übungsblätter

Reguläre Übungsblätter (1-10)

- ▶ Jeweils drei Aufgaben (10 P. + 5 P. + 5 P.)
- ▶ Automatische Bepunktung (direkt nach Upload)
- ▶ Kriterien: Korrektheit, Code Style, Lines of Code
- ▶ **Einzelabgabe!**



Zum Bestehen müssen pro Woche ≥ 8 von 20 Punkten (2 Strikes) und insgesamt ≥ 140 von 200 Punkten erzielt werden.

Ab und an gibt es auch Bonuspunkte (siehe Blatt 0).

Übungen

- ▶ Übungsaufgaben werden in **Git** bzw. **GitLab** gestellt und auch dort abgegeben und bewertet (cf. Schlüsselqualifikation, 1. Semester)
- ▶ 10+1 Übungsblätter

Reguläre Übungsblätter (1-10)

- ▶ Jeweils drei Aufgaben (10 P. + 5 P. + 5 P.)
- ▶ Automatische Bepunktung (direkt nach Upload)
- ▶ Kriterien: Korrektheit, Code Style, Lines of Code
- ▶ **Einzelabgabe!**



Zum Bestehen müssen pro Woche ≥ 8 von 20 Punkten (2 Strikes) und insgesamt ≥ 140 von 200 Punkten erzielt werden.

Ab und an gibt es auch Bonuspunkte (siehe Blatt 0).

Extrablatt 11

- ▶ Hinführung auf erfolgreiches Projekt
- ▶ Keine Bepunktung, aber Feedback

Übungen

- ▶ Übungsaufgaben werden in **Git** bzw. **GitLab** gestellt und auch dort abgegeben und bewertet (cf. Schlüsselqualifikation, 1. Semester)
- ▶ 10+1 Übungsblätter
 - Ausgabe von Übungsblatt i nach VL $i + 1$
 - Abgabe der Lösungen bis 11:59 Uhr am Tag von VL $i + 2$

Tutorien

► Inhalt

- Tutor:innen stellen **Lösungen** vor
- Studierende stellen Fragen zu den eigenen Lösungen
- Besprechen und Vertiefen von Konzepten, die im Programmierkurs nicht ganz klar geworden sind.
- Die **Initiative** geht von Ihnen aus!

► Wichtig!

- Voraussetzung: Intensive Auseinandersetzung mit Stoff und Übungsblatt **vor** dem Tutorium
- Die Übungen lassen sich **nicht** im Rahmen der Tutorien bearbeiten

► Individualisierte Studieneingangsphase (ISE)

- Programmierwerkstatt 1 (INF-21420)
Mittwoch, 13:30–15:00 Uhr, ZT 702
- Zusatzübung Programmieren (INF-21500)
Mittwoch, 11:45–13:15 Uhr, ZT 702

Abschlussprojekt

Zeigen Sie, was Sie gelernt haben!

Abschlussprojekt

Zeigen Sie, was Sie gelernt haben!

Projektablauf

- ▶ Eigenes Projekt zum Thema Schach
- ▶ Extra VL mit allen Details am 09.01.
- ▶ Konzeption/Entwurf im Extrablatt
- ▶ Umsetzung nach Ende der Vorlesungszeit
- ▶ Abgabe: Code, Report, kurzes Video

Für das Projekt werden 100 Punkte vergeben, von denen ≥ 60 Punkte erreicht werden müssen.



Anmeldung – Wichtig

- ▶ Anmeldung für die **Veranstaltung** in ZEuS
 - Teilnahme am Übungsbetrieb
 - Frist: bis zum **Montag, 30. Oktober 2023**

- ▶ Anmeldung für die **Prüfung** in ZEuS
 - Erwerb von ECTS-Punkten
 - First: vom **1. Dezember 2023** bis zum **15. Januar 2024**

GitLab Demo

- ▶ Der Übungsbetrieb wird über GitLab abgewickelt
 - Ausgabe der Übungsblätter als Markdown
 - Abgabe und automatische Bepunktung Ihrer Lösungen
 - **Melden Sie sich is spätestens 30.10. in ZEuS und GitLab an um ein Repo zu erhalten!**
 - `https://gitlab.inf.uni-konstanz.de/ag-storandt/pk1-23/main`