Programming in C/C++

- Introduction -

Organization

Who are we?

Timo Sachsenberg

Lukas Ruppert

Kyowon Jeong

Ayesha Feroz

Arslan Siraj

Faezeh Zakeri

Matteo Pilz

Materials

We will:

- manage lecture materials and information (lecture slides and videos) on **ILIAS.**
- make exercises available on InfoMark.

ILIAS:

- Lecture **slides** will be made available in **ILIAS** before the lecture
- Lecture held via **Zoom**
- Recorded **video** will be made available shortly after the lecture

About **InfoMark** System:

Automatic exercise grading system

- Definition of exercise groups
- Download of new exercise sheets
- Handing in your solutions (code)
- Automated testing of your solution

Organization and testing

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Wed to Fri 05.10.-07.10. and Wed to Fri 12.10.-14.10.
- 11 Lectures (8:30-10:00 and 13:30-15:00)
  - First lectures are a bit longer compared to the later ones

- Exercises
  - Basically, all time when there is no lecture
  - Sheet 0
  - 11 Sheets in total
    - Default: Hand in the next day at noon
    - An extended one from Fri to Wed

- 6 ECTS

- Grade will be made based on your score in the exercises
  - **50% in total to pass**
  - **10% on each sheet to pass**

Practical Exercises

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Zoom rooms for exercises open when there is no lecture

- Right after each lecture in exercise room:
  - Central discussion of the next exercise sheet
  - Breakout rooms to meet with the TAs
    - Chat
    - Screen sharing to discuss individual issues
  - Beware, we might have more than 100 participants vs. 7 TAs (not all of them present every time)
  - Further breakout rooms can be used for peer-to-peer support
    - If you mastered an exercise, please support others
    - Helping others to find a solution is fine but don't share your solution

Practical Exercises – Sheet 0

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- If you have not done Sheet 0 yet, be quick because Exercise 1 (Sheet 1) will start soon!

- Software (see Sheet 0 for details)
  - Linux
  - Clang – compiler
  - CMake – build system generator
  - Ninja – make tool
  - gdb – debugger
  - Visual Studio Code – editor and programming environment

- Setup your account in **InfoMark**

InfoMark

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Registration: Teaming up in groups of 2
- Download exercise sheets (PDF & Code)
- Upload of solution
- Automatic Correction
  - System will not accept any “solution” that does not compile
    - **You will earn 0 points on this task**
    - You can upload as often as you want
  - InfoMark runs public test and gives (immediate) feedback
  - InfoMark also runs (hidden) private tests in order to check if and how well the exercise has been solved
  - TAs will grade based on the InfoMark summary and only occasionally look into your code
  - Please don't print excessive output as this can crash InfoMark and requires us to manually restart the server

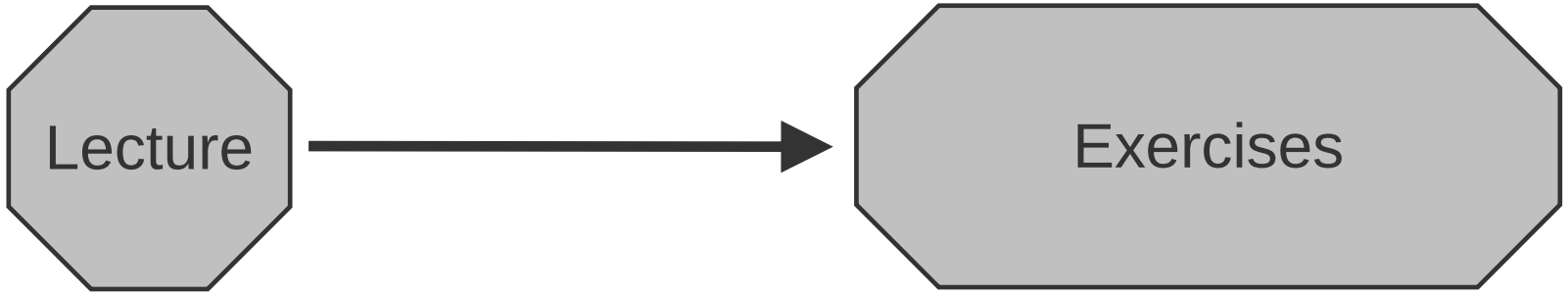
Plagiarism

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- System will check for (exact) copies
- If your group is caught once, you lose most points on that exercise sheet and need to answer to the TAs
- If your group is caught twice, you will automatically fail the course!
- Put your matriculation in each file

Lecture and exercises

- The lecture is supplemented by **practical exercises** that aim at deepening and practicing the lecture material.



- C/C++ Language concepts
- Data structures and Algorithms
- Sample code

- Application of lecture content in exercises
- Gain practical skills using C/C++
- Discussion of exercise sheets

- You only learn C++ programming by writing C++ code yourself and
- by reading and understanding code that others wrote (most code is written by others ;)

What this Course is / is not

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- It is not an introductory course to programming

- You should bring basic programming skills in a higher programming language, e.g., you should have passed Informatik I+II or equivalent

- It is an introduction to the C/C++ language:
  - Focus on **C++**
  - Overview over many language **concepts**, **techniques** and **pitfalls**
  - Newer language features that **already** have good compiler support

This course builds upon material from other courses:
Thanks to Prof. Lensch, Dr. Bielow and Dr. Hauswedel (FU Berlin), and Prof. Kohlbacher for the permission to incorporate their material.

Questions?