

## Course assistants

---

- Course assistants
  - Hannu Kärnä -- first\_name.last\_name@helsinki.fi
- Contact via e-mail or the course telegram channel
- Telegram channel <https://t.me/tktiralabra/> (will contain discourse in Finnish)
- Course page [https://tiralabra.github.io/2022\\_p3/en](https://tiralabra.github.io/2022_p3/en)

## What?

---

- On the course you will implement some program that utilizes "difficult" algorithm(s)/data structure(s) including all of the required data structures and algorithms. Everything included in the prerequisites is considered trivial. Eg. a purely brute force sudoku solver is not a suitable topic.

## Prerequisite knowledge

---

- Prerequisite knowledge:
  - **Data Structures and Algorithms**
    - Mandatory
  - **Programming technique** (only in Finnish)
    - Very beneficial but the course materials cover the required knowledge in testing and project structures
- Ask if:
  - You have done Data Structure and Algorithms but do not have the credit (missing the exam for example)
  - You are unsure about the prerequisite knowledge requirements

## Course content

---

- The programming language to be used has to be accepted by a course assistant. At least Java and Python will be accepted
  - Ask a course assistant if you prefer some other languages
  - Note that testing and test coverage reporting is required no matter the language
- **Git** version control and **GitHub** is used
- The lab consists of **individual work**
- Generally the product will be some sort of **running program**
  - Not a library or a bunch of code that can not be executed
- Typically programs need to have some sort of **user interface**

## Course content

---

- Project examples:
  - Comparison of path finding algorithms
  - Data compression algorithms
  - Cryptography
  - Computational creativity: music, text, pictures
  - Game solvers (minesweepers, nonograms, solitaire...)
    - There is a minesweeper template project on git!
  - AI (Chess, Go...)
    - There is a chess template project on git!
- **Try to choose a topic you are personally interested in!**

## Coding style

---

- The code written for this course should be of high quality and easy to read. You should use some kind of style checking (e.g. Java Checkstyle)
- The project name should be indicative of the contents. Course assistants will not appreciate it if all the project names are along the lines of "Lab-2020"
- Project structure along the lines of the programming techniques course
  - I.e. not all of the project code should be in the same file/folder
- Good coding conventions like DRY ja Single responsibility should be applied

## Conduct of the course

---

- **Deadlines** according to the course material
  - Each deadline gives 0-2 points based on the deadline requirements
  - Submissions are done by *pushing* the project state to GitHub
    - No submissions by e-mail
  - A large part of the points -- and the grade -- are based on deadlines and code reviews
  - After each deadline, a course assistant will give some sort of feedback -- more thorough feedback is available through paja, e-mail or Telegram
  - Extra time for deadlines may be available with **good reasons** if asked for **in advance**.

# Conduct of the course

---

- If you need help, contact the instructor. Personal guidance available in Zoom.
- You may get advice in Telegram too

# Conduct of the course

---

- A **code review** is done in conjunction with Deadlines 4 and 5
  - Every student will get another student project to review
  - Students write and receive feedback on each others' projects
  - The intention is to get familiar with reading code written by others
  - It is also important to get feedback
  - Maximum points for each review is **4**
  - Requirements for grade 5 include making both reviews

# Conduct of the course

---

- At the end of the course there will be a mandatory **demo session**
  - Each student presents their project for about 10 minutes
  - Every student is present for the entire session
  - The project does not have to be completely done at the demo session
- There is **no course exam**

# Motivation

---

- This can be one of the most fun courses during Bachelors' studies - you can implement whatever you want!
- If you get stuck, ask a course assistant for help
  - I'm here for you!
- Normally there are no real penalties for dropping courses – labs are an exception to this
  - It may be harder to enroll to the course after dropping the course

# Ad break

---

- During fall of 2019 a group of students created 2 new project templates for the lab
  - Chess and
  - Minesweeper
- If either of these subjects are of interest, you may want to check them out. Links can be found on the course page

# Thank you!

---

- **Welcome to the course!**
- Most information about the course can be found at: [https://tiralabra.github.io/2022\\_p3/en](https://tiralabra.github.io/2022_p3/en)
  - It's a good idea to read through the entire site! **The English course instructions are not up to date. You should read the Finnish course page too.**