

Distributed Systems

DAT520 - Spring 2026

Course Information

Prof. Hein Meling



Who?

About us

How?

Logistics and Policies...

Where?

Study Resources

Why?

Intro to Distributed Systems (DS)

What?

Syllabus - Main topics

About us

Course Instructor: Prof. Hein Meling

- hein.meling@uis.no
 - Don't send messages on Canvas
- Office: KE D427 (no office hours)
 - Talk to me between lectures
 - Come by the office — I may be busy ...
 - Send email to book an appointment
- Research interests: Distributed Systems
 - Fault tolerance, availability, reliable storage, blockchain systems, etc...

Teaching Assistants

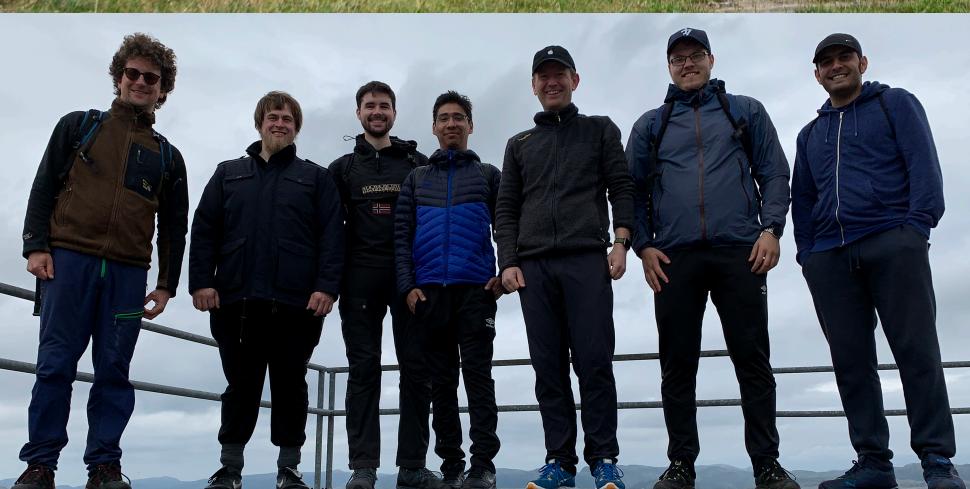
- Daniel Osmundsen Dirdal
- Jostein Lindhom
- Me





{Resilient | Reliable} Systems Lab





Who are you?

Who Are You?

- What's your study program?
 - Computer Science
 - Data Science
 - Others?
- Who is working as a developer?
 - What do you do?
- Experience:
 - Developing distributed systems?
 - Go/Rust/Java/C# programming?
 - Only Python?

Learning Outcomes

Knowledge

Be familiar with

- principles for design and implementation of distributed systems.
- system architectures for distributed systems.
- techniques for solving various problems in distributed systems.

Lectures

Lectures

Lectures

Skills

Be able to

- develop advanced distributed applications with fault tolerance properties.
- perform independent research in distributed systems.
- reason about problems that involve distributed components.

Labs

Master thesis

Exam

General competency

Know how to

- develop distributed computer systems.

Work

Logistics and Policies

Recommended prerequisites

- Operating Systems
- Databases
- Computer Networking
- Advanced Programming
- Discrete Mathematics (Logic operators)
 - For all, existence, set operators, ...

GitHub

<https://dat520.github.io/#/>

- Course information
- Syllabus
- Lecture Plan
- Lab project information
- Reading material (links)

Discord (Helpbot)

- Announcements
- Get help during labs
- Queue management
- Lab approval

<https://discord.gg/ARmwuane>

Register with Helpbot using your GitHub username: /register

PS: No Canvas!

QuickFeed

- Submit assignments to GitHub
- Tested by QuickFeed
- View results of lab
- Lab approval

<https://uis.itest.run/>

*Sign up using your
GitHub account*

Enroll in DAT520

*Consider creating a separate
GitHub account for course work*

Weekly Schedule

| | Mon 5/1 ↔ | Tue 6/1 ↔ | Wed 7/1 ↔ | Thu 8/1 ↔ |
|-------|--|--|------------|---|
| 08:00 | | | | |
| 09:00 | | | | |
| 10:00 | | | | |
| 11:00 | | <div>Laboratorium Lab 10:15-12:00 Room: KE E-456 Staff: D.O. Dirdal</div> | | <div>Laboratorium Lab 10:15-12:00 Room: KE E-456 Staff: D.O. Dirdal</div> |
| 12:00 | <div>Forelesning Lecture 12:15-14:00 Room: KE E-456 Staff: H. Meling</div> | <div>Forelesning Lecture 12:15-14:00 Room: KE E-456 Staff: H. Meling</div> | | |
| 13:00 | | | | |
| 14:00 | | | | |

Lecture Plan Spring 2026

| Week | Month | Monday (12:15 - 14:00) | Tuesday (12:15 - 14:00) | Comment |
|------|-------|--------------------------|-----------------------------|------------------|
| 2 | Jan | Course Introduction | Ch 1 Introduction | |
| 3 | Jan | Live coding: Go / gRPC | Live coding: Gorums | Bring laptop |
| 4 | Jan | Live coding: Gorums | Ch 2 Basic Abstractions | Bring laptop |
| 5 | Jan | Ch 2 Basic Abstractions | Ch 3 Reliable Broadcast | |
| 6 | Feb | Ch 3 Reliable Broadcast | Paxos | |
| 7 | Feb | Paxos | Ch 5 Consensus | |
| 8 | Feb | Ch 6 Consensus Variants | Ch 6 Consensus Variants | |
| 9 | Feb | Ch 4 Shared Memory | Ch 4 Shared Memory | |
| 10 | Mar | Consistent Global States | Open Day: No Lecture or Lab | |
| 11 | Mar | Consistent Global States | Consistent Global States | |
| 12 | Mar | Guest Lecture | | |
| 13 | Mar | No Lecture (Only Lab) | No Lecture (Only Lab) | |
| 14 | Apr | Easter: No Lab | Easter: No Lab | Work on your own |
| 15 | Apr | No Lecture (Only Lab) | No Lecture (Only Lab) | |
| | Apr | | | |
| 17 | Apr | No Lecture (Only Lab) | No Lecture (Only Lab) | |
| 22 | Jun | Written exam: TBD | | |

“I love deadlines. I like the whooshing sound
they make as they fly by.”

— Douglas Adams, The Salmon of Doubt

Lab Assignments

| Lab | Topic | Grading | Approval | Submission | Effort | Deadline |
|-----|--|------------|-------------|--------------|-------------|-------------|
| 1 | <u>Getting Started with Network Programming</u> | Pass/fail | Automatic | Individually | 2-3 hours | January 15 |
| 2 | <u>Network Programming with gRPC and Gorums</u> | Pass/fail | Automatic | Individually | 5-8 hours | January 22 |
| 3 | <u>Failure Detector and Leader Election</u> | Pass/fail | TA Approval | Group | 20-25 hours | February 12 |
| 4 | <u>Single-decree Paxos and Multi-Paxos</u> | Pass/fail | TA Approval | Group | 30-40 hours | March 5 |
| 5 | <u>Multi-Paxos with Gorums</u> | Pass/fail | TA Approval | Group | 20-30 hours | March 19 |
| 6 | <u>Multi-Paxos with Clients and Performance Evaluation</u> | Pass/fail | TA Approval | Group | 20-30 hours | April 9 |
| 7 | <u>Replicated Bank Application</u> | No grading | No Approval | Group | 20-30 hours | April 23 |

Deadlines and Slip days

dat520.github.io/#/policy

- If you cannot make a deadline, you can use up to **7 slip days** throughout the semester, including weekends and holidays.
- It is the **date on your lab's submission**, as viewed on GitHub that counts towards the slip days.
- Slip days **cannot be used to extend** the final submission deadline.
- All submissions must be approved before the last deadline.

Collaboration

dat520.github.io/#/policy

- Talk to each other or the teaching staff about any of the assignments. Assistance is limited to general discussion of the problem. Each student/group must write out their own solutions.
- Consulting another student's/group's solution is prohibited. Submitted solutions must not be copied from any source.
- You **shall not** supply your work to other students in future instances of this course.



Code Reuse and Distribution

dat520.github.io/#/policy

- You are prohibited from publishing, sharing, or redistributing your solutions in any form, including but not limited to:
 - Public repositories on GitHub, GitLab, Bitbucket, or similar platforms
 - Code-sharing websites such as Chegg, Course Hero, or Pastebin
 - Personal blogs, websites, or social media



Generative Models

dat520.github.io/#/policy

- May use generative models such as Claude or GitHub Copilot to generate code.
- Must be able to explain the code as if you had written it yourself.
- The code must obviously **solve the assignment** and pass a sufficient number of tests on QuickFeed.



Generative Models

dat520.github.io/#/policy

- **Adaptation:** Review and adapt any generated code to ensure correctness, efficiency, and alignment with the assignment.
- **Improvement:** Refine generated code to follow current best practices, modern idioms, and up-to-date language features.
- **Accountability:** You are fully responsible for all submitted code, whether AI-generated or written by hand.



Contributions

dat520.github.io/#/policy

- Group members must **contribute equally** to the code
- *Implies that group members should commit a similar amount of code on GitHub.*
- Each group member must **present individually** and explain their work to a TA for approval.
- Group members are expected to be able to explain the code written by other group members.



Two Working Styles

dat520.github.io/#/policy

- Pair programming
 - Alternate between coding vs watching
 - Make sure to switch user that writes and commits code to GitHub
- Code Review
 - Each group member creates a branch
 - Create Pull Request
 - Partner reviews your code before it is merged into the main branch

When to Commit

dat520.github.io/#/policy

- Commits should represent a **logical unit of work**.
 - A bug fix, a new feature, a refactoring, etc.
 - It is up to you to decide what constitutes a unit of work, but it should not be too large.
- *Unacceptable to commit an entire lab assignment in one commit.*

Commit Messages

dat520.github.io/#/policy

- Describe the changes you have made.
- Must follow the Conventional Commits format.

| | |
|---|---|
| 93fa221 Hein Meling Mon Jan 8 13:53:59 2024 +0100 | fix(lab1): fixed doc format issue in errors task |
| 90509b3 Hein Meling Mon Jan 8 13:51:13 2024 +0100 | feat(lab1): implemented fibonacci sequence task |
| 43a353c Hein Meling Mon Jan 8 13:49:57 2024 +0100 | feat(lab1): implemented multiwriter task |
| 9d5d0bf Hein Meling Mon Jan 8 13:49:39 2024 +0100 | feat(lab1): implemented errors task |
| 2693212 Hein Meling Mon Jan 8 13:44:50 2024 +0100 | feat(lab1): implemented the cipher task |
| f6d8a83 Hein Meling Mon Jan 8 13:43:12 2024 +0100 | feat(lab1): implemented stringer task |
| 1f9a64a Hein Meling Mon Jan 8 13:40:55 2024 +0100 | mc-go(lab1): answered go questions |
| 88da89b Hein Meling Mon Jan 8 13:38:25 2024 +0100 | token(lab1): ran generate_token on go1 |
| 23e522c Hein Meling Mon Jan 8 13:20:26 2024 +0100 | mc-missing(lab1): answered missing semester questions |
| 24cf9ac Hein Meling Mon Jan 8 13:19:01 2024 +0100 | mc-shell(lab1): answered shell questions |
| 31c4fa8 Hein Meling Mon Jan 8 13:17:54 2024 +0100 | mc-git(lab1): fixed git answers |
| ce0027e Hein Meling Mon Jan 8 11:19:53 2024 +0100 | mc-git(lab1): answered git questions |
| 8133f91 Hein Meling Mon Jan 8 09:52:23 2024 +0100 | lab1: first commit |

Tracking Your Time

dat520.github.io/#/policy

- The assignments repository has a hours.csv file

| Column | Required | Format | Description |
|----------|----------|-------------|--|
| lab | Yes | lab1 - lab8 | Lab identifier matching folder names |
| student | Yes | Text | Your name (use consistent spelling) |
| date | Yes | YYYY-MM-DD | Date of work session |
| hours | Yes | Decimal | Hours spent (e.g., 1.5 , 2.0) |
| category | Yes | Enum | One of: reading , coding , debugging , testing , other |
| notes | Yes | Text | Description of work done |

Important: Use a dot (.) as the decimal separator for hours, not a comma (,). For example, write 2.5 for two and a half hours, not 2,5 .

Tracking Your Time

dat520.github.io/#/policy

- The assignments repository has a hours.csv file
- Example

```
lab,student,date,hours,category,notes
lab1,Alice,2025-01-15,2.5,reading,"Read UDP documentation"
lab1,Alice,2025-01-16,3.0,coding,"Implemented echo server"
lab1,Bob,2025-01-15,1.5,reading,"Studied Go networking"
lab1,Bob,2025-01-17,4.0,debugging,"Fixed UDP client issues"
```

Group Registration on QuickFeed

dat520.github.io/#/policy

- Groups of two (max three)
- If you cannot find a group partner
 - Ask on the #group-maker channel on Discord

Partner Problems

dat520.github.io/#/policy

- Please **tell us** and **your partner** early
 - decide to drop out of the course
 - want to dissolve the group and work alone
- To avoid problems, we suggest to organize regular
 - joint work sessions and
 - coordination meetings

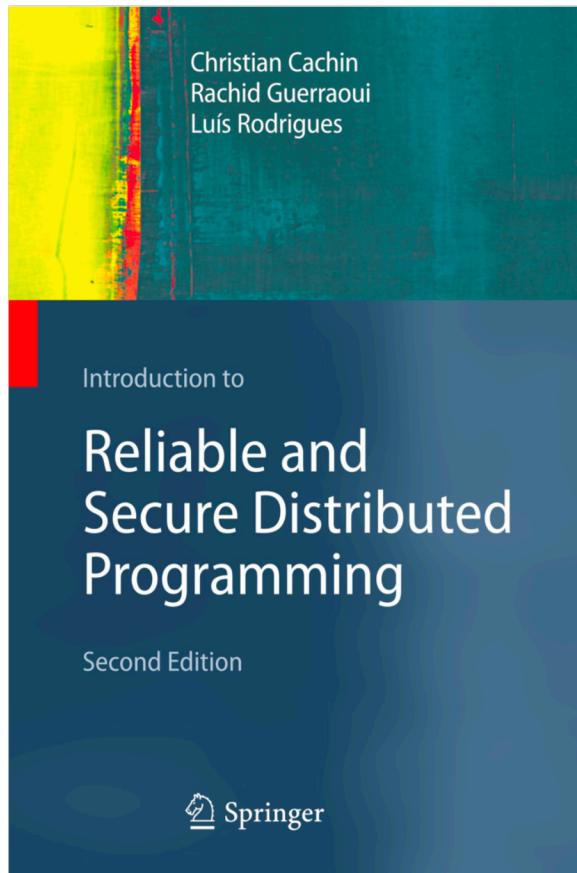


Tell us!

Study Resources

Official Book

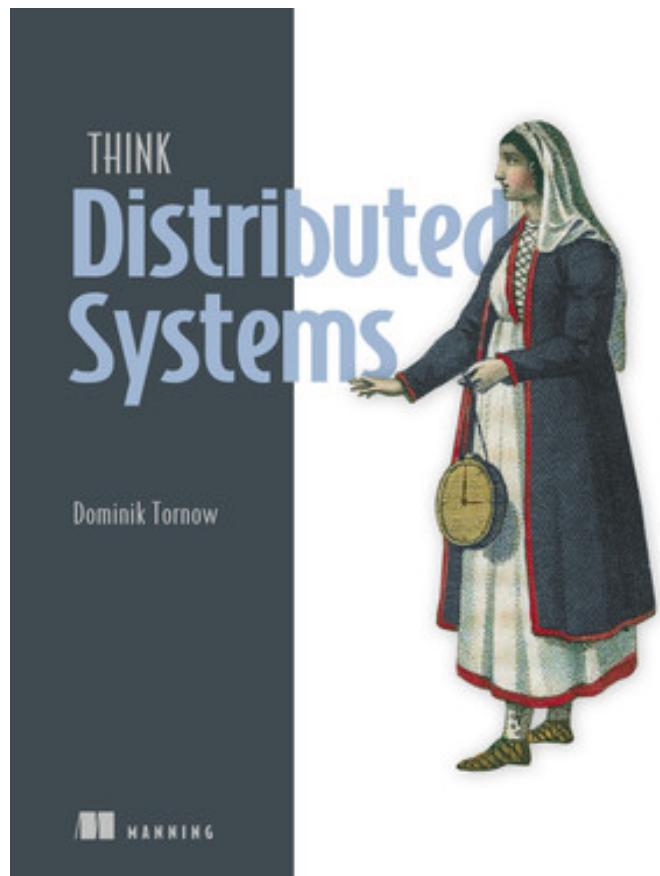
<https://link.springer.com/book/10.1007%2F978-3-642-15260-3>



- Available for free download when on UiS campus
- Available in print in the bookstore
- RSDP

Optional Book

<https://www.oreilly.com/library/view/think-distributed-systems/9781633436176/>



- September 2025
- May be easier to read, has more illustrations
- Should be available in print in the bookstore

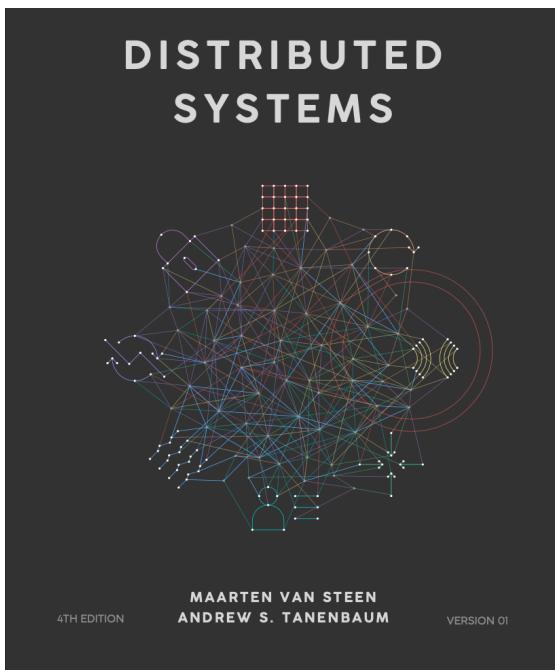
dat520.github.io/#/syllabus

- Chapters from RSDP
- Additional papers
- Lab project
- Lecture slides

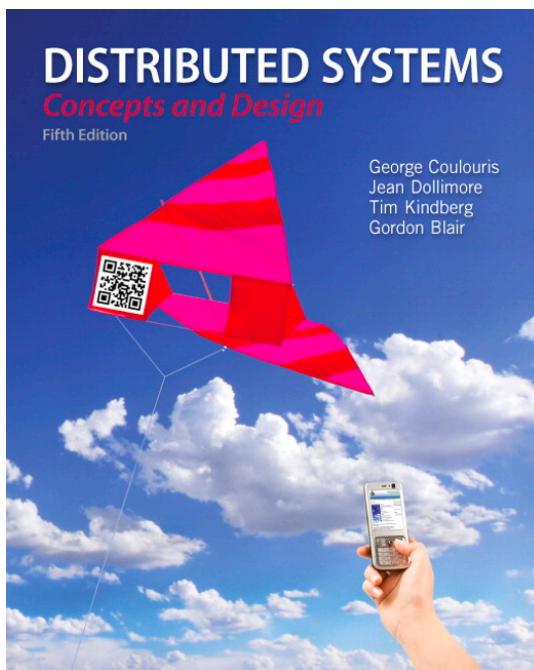
dat520.github.io/#/syllabus

- Lab exercises includes some theory
- Theory exercises and solutions in textbook
 - Recommended practice
 - Exam questions may be picked from textbook exercises
- Previous exams
 - Will post on Canvas

Other Books



Available free online



Print copy available
at UiS library

Links

- Official web page: go.dev
- [A Tour of Go](#)
- [Effective Go](#)
- Other [Learning resources](#)
- [Standard library](#) and [Style guide](#)
- Video: [Go Tutorial](#) from freeCodeCamp.org (7 hours)
 - Beware: some information may be outdated!!



Alan A. A. Donovan
Brian W. Kernighan



Questions?