

Practical Information

Lecture 0 of TDA384/DIT391

Principles of Concurrent Programming

Nir Piterman

Chalmers University of Technology | University of Gothenburg

SP1 2021/2022



UNIVERSITY OF
GOTHENBURG



CHALMERS
UNIVERSITY OF TECHNOLOGY

Canvas Room and Course Website

Make sure to regularly check the **Canvas Room** and **Course Website**:

Canvas Announcements, discussion forum, videos

CTH login <https://canvas.chalmers.se/courses/14929>

GU login <https://canvas.gu.se/courses/12523~14929>

Website Lectures, labs, exams, ...

http://www.cse.chalmers.se/edu/course/TDA384_LP1

These should be your primary sources of information about the course.

Discussion Forum

Use the **Canvas discussion forum** for questions and discussions of general interest to the course:

https://canvas.chalmers.se/courses/14929/discussion_topics

https://canvas.gu.se/courses/12523~14929/discussion_topics

The forum URL is of course linked from the course website.

**Do not share solutions to labs on Canvas
(or anywhere else) !!!**



Covid 19 – The course is running mostly virtually!

- Lectures:

- Some lectures are given on campus.
 - Well done! You are here!
- All lectures are broadcast / running on zoom:
- <https://chalmers.zoom.us/j/63362388303?pwd=dTNuS1NWQWdsOWpRaVJTRTh1Ympxdz09>
- Meeting ID: 63362388303 Password: 268075
- Lectures are recorded:
 - Student video is not captured.
 - If you ask a question your voice will be captured.
 - Videos will be shared through closed Canvas access.
 - After course: either delete or remove all student information.

Covid 19 – The course is running mostly virtually!

- Labs:

- Lab assistance requests –

- Create a Zoom meeting w.o. password
 - Put support requests on [Waglys](#)
 - Name for support request (limited to 20 chars):
 - Zoom meeting ID (not link)
 - Add Chalmers ID (if possible)

- Demo signup –

- A doodle with available slots will be posted on the appropriate lab page before each deadline
 - Create a Zoom meeting (w.o. password)
 - Register the day **before** the demos
 - Use group ID + Zoom meeting ID as name in the poll
 - Be on Zoom 5 minutes before your time and be ready to run the demo

Zoom Etiquette

- Login with your Chalmers/GU account (use SSO login).
- Set your profile's photo.
- Mute your microphone!
- If you can't hear me, write on the chat. Unmute only as last resort.
- If you have a question, raise your hand (virtually!). Ask questions on the chat.
- It is hard to present and manage the meeting at the same time!



The teaching team

Lecturer/Examiner

- Nir Piterman



Teaching assistants (TAs)

- Matthías Páll Gissurarson
- William Hughes
- Agustin Mista



If you have questions

- Ask them during lectures and lab sessions.
- Post them on the [discussion forum](#).
- Send an email to pcp-teachers@lists.chalmers.se (of personal nature)
- Book an appointment with the teacher or TAs (by email).

Protip: options 1 & 2 are quicker than options 3 & 4.

Student Representatives

Chalmers

- You?

GU

- You?

Main Learning Goals

- By the end of the course you should be able to
 - Understand the problems common to concurrent and parallel systems,
 - Demonstrate techniques and patterns to reason about and write correct and efficient concurrent programs,
 - Apply those techniques and patterns in modern programming languages.

Overview of the Course

- Introduction to concurrency.
- **Part 1.** Classic, shared-memory concurrency in Java:
 - java threads,
 - locks, semaphores, and monitors.
- **Part 2.** Message-passing concurrency:
 - Erlang and the actor model.
- **Part 3.** Parallelizing computations:
 - fork/join parallelism,
 - lock-free programming

Lectures

Class #	Date	Topic
1	Mon, 30 Aug 2021	Introduction to concurrent programming
2	Mon, 30 Aug 2021	Races, locks, and semaphores
3	Thu, 2 Sep 2021	Models of Concurrency and Synchronization Algorithms
4	Fri, 3 Sep 2021	Java Tutorial
5	Mon, 6 Sep 2021	Synchronization Problems with Semaphores
6	Thu, 9 Sep 2021	Monitors
7	Fri, 10 Sep 2021	Introduction to Functional Programming in Erlang
8	Mon, 13 Sep 2021	Message-passing concurrency in Erlang
9	Wed, 15 Sep 2021	Synchronization problems with message passing
10	Mon, 20 Sep 2021	Parallelizing computations
11	Thu, 23 Sep 2021	Parallel Linked Lists
12	Fri, 24 Sep 2021	Parallel Queues
13	Mon, 27 Sep 2021	Lock-free programming (STM)
14	Fri, 1 Oct 2021	Verification of Concurrent Programming
15	Mon, 4 Oct 2021	Concurrency in weak memory models (to be moved earlier)
16	Fri, 15 Oct 2021	Invited Lecture (TBD)
17	Mon, 18 Oct 2021	TBD
18	Fri, 22 Oct 2021	Revision and exam preparation

Make sure to check up-to-date schedule [on website!](#)

Labs

There will be four labs – one for setup and one for each part of the course.

0. Setup (Fire, Java, Erlang)
1. Trainspotting (Java)
2. CCHAT (Erlang)
3. A-mazed (Java)

Descriptions of the labs, deadlines, and rules are [on the website](#).

- Register your group (2 students) in [Fire](#).
- Make sure to check the lab schedule on the website.

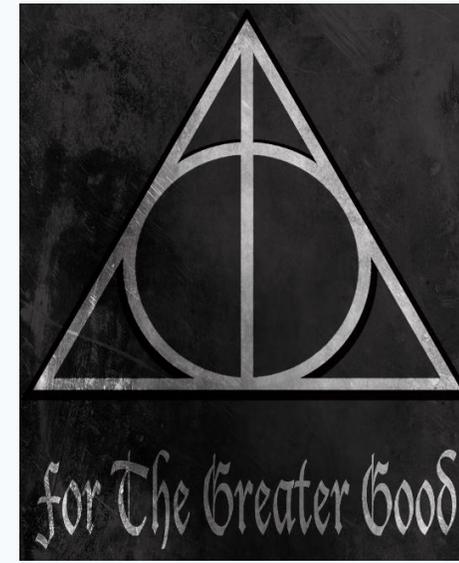
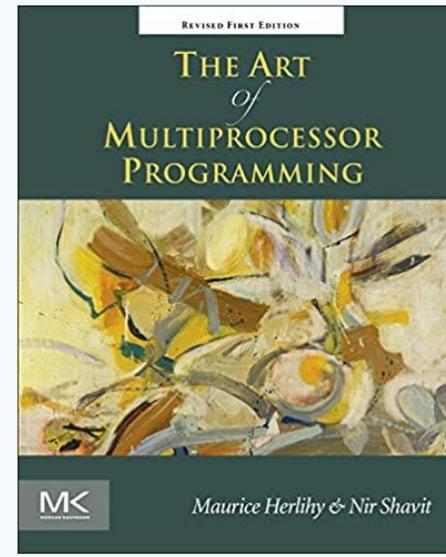
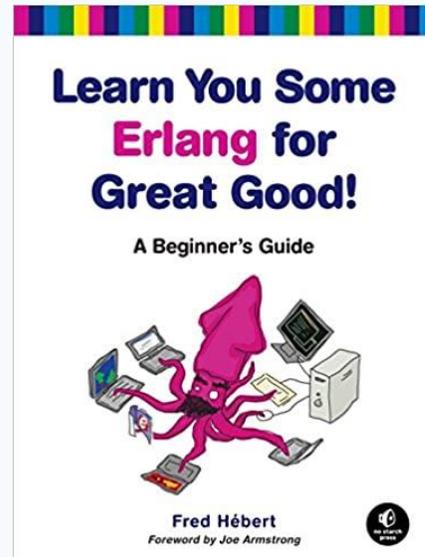
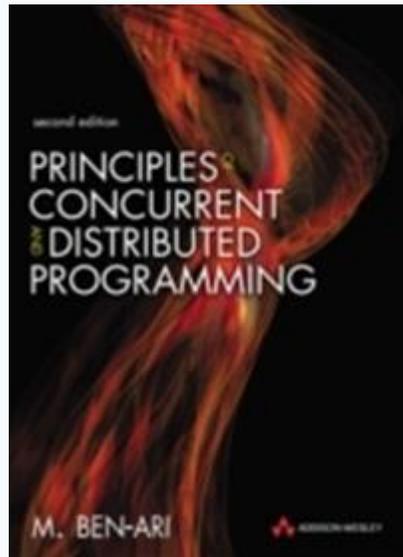
**Do not share solutions to labs on Canvas
(or anywhere else) !!!**

Slides and Reading Material

Lecture slides: will be [on the website](#).

Books:

- Ben-Ari: Principles of concurrent and distributed programming, 2nd edition.
- Hébert: Learn you some Erlang for great good (free online),
- Herlihy & Shavit: The art of multiprocessor programming



Exam

It is currently not known whether exams will be held online / on campus.

- Open-book exam:
 - max. 2 textbooks,
 - max. 4 two-sided A4 sheets of notes (printed or handwritten),
 - an English dictionary.
- All topics in the lectures can be examined (except the guest lecture).
- See exams of previous years for examples ([on the website](#)).
- Exam dates:
 - 23 October 2021
 - 14 March 2022, 18 August 2022 (re-exam)
- Check the website for updates!
- Exam grading: [see the course website](#).

Computing Resources

- Install Java and Erlang/OTP on your computers.
- Try out the examples presented in class; the complete examples will be available [on the website](#) for each lecture.
- Lab 1 (Trainspotting) requires a simulator, which runs on the lab computers (Unix/Linux workstations).
- See the course website for [instructions](#) on how to
 - use the lab computers, and
 - set up Java & Erlang/OTP on your own computers.



There's a lab on Thursday – What's the point of that?

- Complete the setup assignment!
- Create the groups!
- Setup the train system!
- Start playing with it with sequential programs:
 - Have only one train.
 - Start and stop.
 - Check distances and speeds.
 - When is a train on a switch?
 - Make plans.

Erlang, Erlang, Erlang, ...

- Start early!
- Install the Erlang environment.
- Start the online tutorial.
- Especially if never done functional programming before.
- Compared to previous years:
 - Erlang lab extended to 3.5 weeks
 - First lab support moved to Fri 17 Sep
 - Demo moved to Mon 11 Oct

© 2016–2019 Carlo A. Furia, Sandro Stucki



Except where otherwise noted, this work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.

To view a copy of this license, visit

<http://creativecommons.org/licenses/by-sa/4.0/>.