

Practical Information

Lecture 0 of TDA384/DIT391

Principles of Concurrent Programming

Nir Piterman and **Gerardo Schneider**

Chalmers University of Technology | University of Gothenburg

SP3 2022/2023



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/22478>

GU login <https://canvas.gu.se/courses/12523~22478>

Website Lectures, labs, exams, ...

http://www.cse.chalmers.se/edu/course/TDA384_LP3

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/22478/discussion_topics

https://canvas.gu.se/courses/12523~22478/discussion_topics

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

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

Lectures

- Most lectures are given in **HB1**
- Two lectures in **SB-H2**
- One lecture in **HA1**
- One lecture in **HB3**
- One lecture in **HC4**
- **Check out TimeEdit!**

		Mon	Wed	Fri
		10:00-11:45	10:00-11:45	13:15-15:00
Week 3	16/01-20/01	16 Jan 2023	18 Jan 2023	20 Jan 2023
Week 4	23/01-27/01	23 Jan 2023		27 Jan 2023
Week 5	30/01-03/02		01 Feb 2023	03 Feb 2023
Week 6	06/02-10/02	06 Feb 2023		
Week 7	13/02-17/02	13 Feb 2023	15 Feb 2023	17 Feb 2023
Week 8	20/02-24/02			
Week 9	27/02-03/03			
Week 10	06/03-10/03	06 Mar 2023		

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

Labs

- Mixing physical and online labs
 - Demos and labs on **Fridays at 8:00** will be **online**

CHECK COURSE HOMEPAGE AND ANNOUNCEMENTS!

The teaching team

Lecturer/Examiner

- Gerardo Schneider



Teaching assistants

(TAs)



William Hughes



David Lidell



Luca Di Stefano



Samuel Kyletoft



Martin Jonsson

If you have questions

- Ask them during lectures and lab sessions
- Post them on Canvas [discussion forum](#)
 - Questions of general interest
- Send an email to pcp-teachers@lists.chalmers.se
 - Questions of personal nature / requiring a quick reaction from the teachers
- Book an appointment with the teacher or TAs (by email to pcp-teachers)

Student Representatives

Chalmers

- Johan Berg (johanberg2001@gmail.com)
TKITE
- Nils Ekström (venomous.steam@gmail.com)
TKDAT
- Binsha Nazar (binshanazaron@gmail.com)
MPHPC
- Aileen Vergara Abdali (aileenv@student.chalmers.se)
TKDAT
- Johan Ödesjö (Johan.odesjo@gmail.com)
TKTEM

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
 - In practice, focus on Java and Erlang

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

Class #	Date	Topic
1	Mon, Jan 16	Introduction to the course
2	Wed, Jan 18	Races, locks and semaphores
3	Wed, Jan 18	Models of concurrency and synchronization algorithms
3	Fri, Jan 20	Models of concurrency and synchronization algorithms (Cont.)
4	Mon, Jan 23	Synchronization problems with semaphores
5	Wed, Jan 25	Monitors
5	Fri, Jan 27	Introduction to functional programming in Erlang
6	Wed, Feb 1	Message-passing concurrency in Erlang
7	Fri, Feb 3	Synchronization problems with message-passing
8	Mon, Feb 6	Parallelizing computations
9	Mon, Feb 13	Parallel linked lists
9	Wed, Feb 15	Lock-free programming
10	Wed, Feb 15	Verification of concurrent programs
11	Fri, Feb 17	Guest lecture (TBA)
13	Mon, Mar 6	Revision

Lectures

- 12 lectures + 1 revision lecture + 1 guest lecture (TBA)
- 2 Tutorials (Java and Erlang)
- Some lectures will take less/more time -> the schema might then be rescheduled

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

Labs

There will be one preparation lab and three “real” labs – one for each part of the course:

1. Trainspotting (Java)
2. CCHAT (Erlang)
3. A-mazed (Java)

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

- **Lab 0:** Set up and register your group (2 students) in [Fire](#)
- Make sure to check the lab/room schedule on the website

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

Tutorials

There will be 2 tutorials

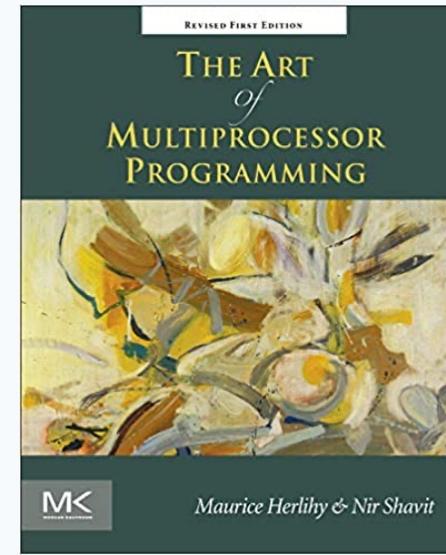
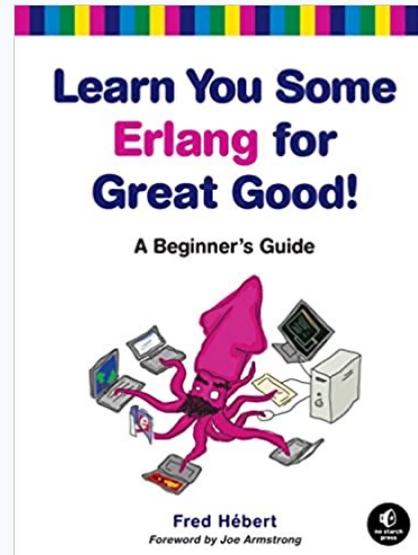
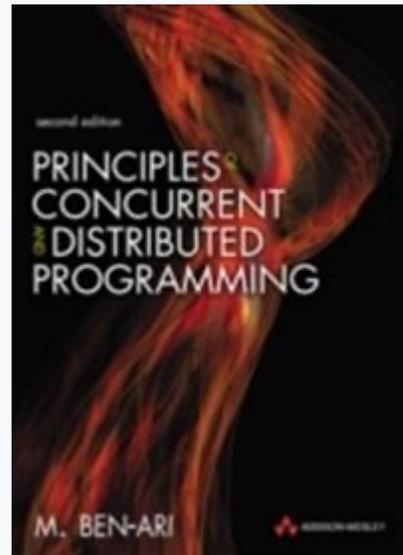
1. **Java** tutorial: **Wed Jan 18 at 15:15**
2. **Erlang** tutorial: **Fri Feb 3 at 08:00**

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

- 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 guest lectures)
- See exams of previous years for examples ([on the website](#))
- Exam dates:
 - **13 March 2023**
 - **17 August 2023** (re-exam)
 - **October 2023** (TBA)
- 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 are two lab sessions this week (Wed and Fri) – What's the point of that?

- LAB 0: Setup the system, create your group, etc.
- 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
- And of course, the **Java Tutorial**

Check the webpage regularly!

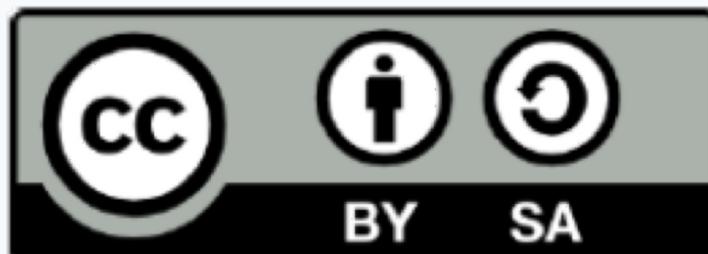
Erlang, Erlang, Erlang, ...

- Start early!
- Install the Erlang environment
- Start the online tutorial
- Especially if you have never done functional programming before
- Attend the **Erlang Tutorial!**

Course Evaluation

- Please remember to fill in the course evaluation (“*kursvärdering*”) when the time comes!
 - Important feedback for us
 - To know what can be improved as well as what is working well

© 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/>.