

Practical Information

Lecture 0 of TDA384/DIT391

Principles of Concurrent Programming

Nir Piterman and Gerardo Schneider

Chalmers University of Technology | University of Gothenburg

SP1 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/20098>

GU login <https://canvas.gu.se/courses/12523~20098>

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/20098/discussion_topics

https://canvas.gu.se/courses/12523~20098/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

- Check out TimeEdit.
- All lectures are given in HC2
 - Well done! You are here!

TDA384 PCP Schedule HT22 : Lectures					
		Mon 15:15-17:0	Wed	Thu 08:00-09:45	Fri 15:15-17:00
Week 35	29/8-2/9		31/8 13:15-17:00	1 Sep	2 Sep
Week 36	5-9/9	5 Sep		8 Sep	9 Sep
Week 37	12-16/9	12 Sep	14/9 08:00-09:45		
Week 38	19-23/9	19 Sep		22 Sep	23 Sep
Week 39	26-30/9	26 Sep			30 Sep
Week 40	3-7/10	3 Oct			
Week 41	10-14/10				14 Oct
Week 42	17-21/10	17 Oct			21 Oct

Lectures
>
<

Labs

- Mixing physical and online 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

The teaching team

Lecturer/Examiner

- Nir Piterman



Teaching assistants (TAs)

- William Hughes
- Agustin Mista
- Abhiroop Sarkar



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

- Pegah Amanzadeh
pegahamanzadeh@gmail.com
TKITE
- Edward Karlsson
edward1337@live.com
MPSOF
- Samuel Kyletoft
samuel@kyletoft.se
TKDAT
- Enayatullah Norozi
norozie@student.chalmers.se
TKDAT
- Hugo Stegrell
sthugo@student.chalmers.se
MPCSN

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

- 14 lectures + 1 revision lecture
- 2 Tutorials (Java and Erlang)
- Subject to change
- Some lectures will take less/more time -> the schema might then be rescheduled

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

Class #	Date	Topic
1	Wed, Aug 31	Introduction to concurrent programming
2	Wed, Aug 31	Races locks and semaphores
3	Thu, Sep 1	Models of concurrency and synchronization algorithms
4	Fri, Sep 2	Java Tutorial
5	Mon, Sep 5	Synchronization problems with semaphores
6	Thu, Sep 8	Monitors
7	Fri, Sep 9	Intro to FP in Erlang
8	Mon, Sep 12	Message-passing concurrency in Erlang
9	Wed, Sep 14	Sync problems with message-passing
10	Mon, Sep 19	Parallelizing computations
11	Thu, Sep 22	Parallel linked lists
12	Fri, Sep 23	Parallel Queues
13	Mon, Sep 26	Lock-free programming (STM)
14	Fri, Sep 30	Verification of Concurrent Programming
15	Mon, Oct 3	Concurrency in Weak Memory Models
16	Fri, Oct 14	TBD
17	Mon, Oct 17	TBD
18	Fri, Oct 21	Revision

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

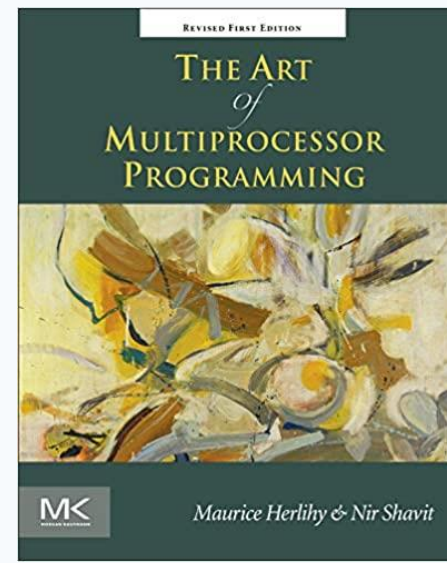
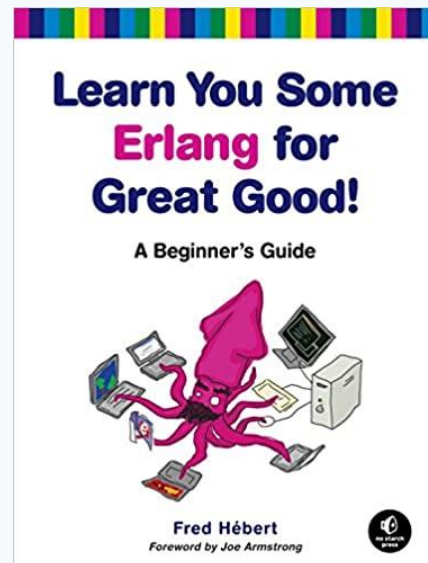
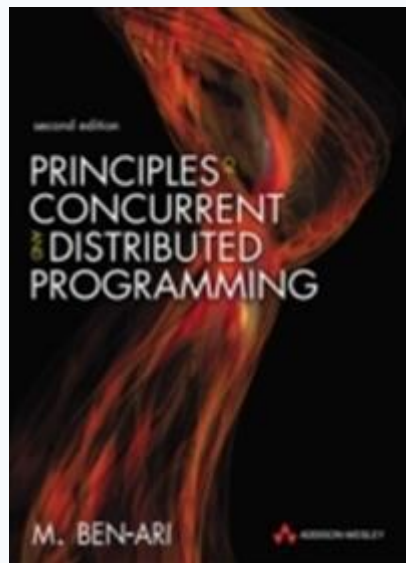
1. **Java** tutorial (today? tomorrow? Friday?)
2. **Erlang** tutorial (Lecture 7 (hopefully next week))

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:
 - **20 October 2022**
 - **13 March 2022, 17 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 tomorrow – 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.
- Attend the **Erlang 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

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/>.