



UNIVERSITY OF  
GOTHENBURG

# Practical information

Lecture 0 of TDA384/DIT391

Principles of Concurrent Programming

---

K V S Prasad

Chalmers University of Technology | University of Gothenburg

SP1 2019/2020

# Canvas room and course website

Make sure to regularly check the **Canvas room** and **course website**:

**Canvas** announcements, discussion forum.

CTH login <https://canvas.chalmers.se/courses/7493/>

GU login <https://canvas.gu.se/courses/12523~7493/>

**Website** lectures, labs, exams, ...

[http://www.cse.chalmers.se/edu/course/TDA384\\_LP1/](http://www.cse.chalmers.se/edu/course/TDA384_LP1/)

These are the 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/7493/discussion\\_topics](https://canvas.chalmers.se/courses/7493/discussion_topics)

[https://canvas.gu.se/courses/12523-7493/discussion\\_topics](https://canvas.gu.se/courses/12523-7493/discussion_topics)

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

# Discussion forum

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

[https://canvas.chalmers.se/courses/7493/discussion\\_topics](https://canvas.chalmers.se/courses/7493/discussion_topics)

[https://canvas.gu.se/courses/12523-7493/discussion\\_topics](https://canvas.gu.se/courses/12523-7493/discussion_topics)

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

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

# The teaching team

## Course responsables

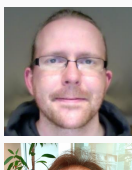
**Examiner** K.V.S. Prasad

**Lecturer** Nir Piterman



## Teaching assistants (TAs)

- Sandro Stucki
- Herbert Lange
- Matthías Páll Gissurarson
- Ken Bäcklund



## If you have questions...

1. ask them during the lectures and lab sessions,
2. post them in the discussion forum **on Canvas**,
3. send an email to **`pcp-teachers@lists.chalmers.se`**,
4. book an appointment with the teacher or TAs (by email).

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

# Student representatives

## Chalmers student representatives

- Johannes Binde, [binde@student.chalmers.se](mailto:binde@student.chalmers.se)
- Thomas Frödin Larsson, [frodin@student.chalmers.se](mailto:frodin@student.chalmers.se)
- David Hedgren, [davhedg@student.chalmers.se](mailto:davhedg@student.chalmers.se)
- Pontus Johansson, [ponjo@student.chalmers.se](mailto:ponjo@student.chalmers.se)
- Axel Karlsson, [axeka@student.chalmers.se](mailto:axeka@student.chalmers.se)
- Markus Pettersson, [markp@student.chalmers.se](mailto:markp@student.chalmers.se)
- Henrik Valter, [valterh@student.chalmers.se](mailto:valterh@student.chalmers.se)

## GU student representatives

- Sophia Pham, [gusphaso@student.gu.se](mailto:gusphaso@student.gu.se)
- Isabella Fransson, [gusisabefr@student.gu.se](mailto:gusisabefr@student.gu.se)
- Johan Berg, [gusberjofx@student.gu.se](mailto:gusberjofx@student.gu.se)
- Mirai Ibrahim, [gusibrmi@student.gu.se](mailto:gusibrmi@student.gu.se)
- Johanna Thall, [gusthalljo@student.gu.se](mailto:gusthalljo@student.gu.se)
- Irja Vuorela, [gusvuoir@student.gu.se](mailto:gusvuoir@student.gu.se)

# 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

---

0	Promela bootcamp	2 September 2019
1	Introduction to concurrent programming	2 September 2019
2	Races, locks, and semaphores	6 September 2019
3	Models of concurrency & synchronization algorithms	9 September 2019
4	Synchronization problems with semaphores	12 September 2018
5	Monitors & Java concurrency tutorial	13 September 2018

---

6	Introduction to functional programming in Erlang	16 September 2018
	Erlang Tutorial	20 September 2019
7	Message-passing concurrency in Erlang	23 September 2019
8	Synchronization problems with message passing	27 September 2019

---

9	Parallelizing computations	30 September 2019
10	Parallel linked lists	4 October 2019
11	Lock free programming	7 October 2019
	Guest lecture	14 October 2019
12	Models and languages of concurrent computation	18 October 2019
13	Verification of concurrent programs & course recap	21 October 2019
14	Formal reasoning examples and proofs & exam prep.	25 October 2019

---

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

There will be three 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](#).

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

There will be three 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](#).

- 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) !!!**

# Slides and reading material

Lecture slides: 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*



- 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:
  - to be announced later,
  - 26 October 2019, 14:00–18:00,
  - 9 January 2020, 14:00–18:00 (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.

## These slides' license

© 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/>.