

Principles of Concurrent programming: practical information

TDA384/DIT391

Carlo A. Furia



CHALMERS



One website to rule them all

Make sure to check out regularly the course website:

http://www.cse.chalmers.se/edu/year/2017/course/TDA384_LP3/

It is the primary source of all information about the course.

Message board

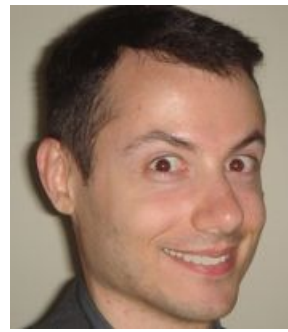
Please sign up in Piazza, and use its message board for questions and discussions of general interest to the course:

<https://piazza.com/chalmers.se/spring2018/tda384dit391/home>

The Piazza URL is of course linked from the course homepage.

The teaching team

Lecturer/examiner: Carlo A. Furia



Teaching assistants (TAs):

- Alexander Sjösten
- Ann Lillieström
- Iulia Bastys
- Mauricio Chimento



Student representatives

- **CTH** student representatives:
 - Magnus Carlsson, magc@student.chalmers.se
 - Denis Furian, furian@student.chalmers.se
 - Mateo Raspudic, raspudic@student.chalmers.se
 - Arnaud Terrailon,
arnaud.terraillon@telecom-bretagne.eu
 - Karl Wikström, karlwik@student.chalmers.se
- **GU** student representatives will be announced later

Main learning goals

- Understanding the problems common to concurrent and parallel systems
- Demonstrating techniques and patterns to write correct and efficient concurrent programs
- Applying the techniques and patterns in modern programming languages

Main topics

- Introduction to concurrency
- Classic concurrency:
 - Java threads
 - Locks, semaphores, and monitors
- Message-passing concurrency:
 - Erlang and the actor model
- Parallelizing computations:
 - Fork/join parallelism
 - Lock-free programming

Lectures

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

| Class # | Date | Topic |
|---------|------------------|--|
| 1 | 15 January 2018 | Introduction to concurrent programming |
| 2 | 15 January 2018 | Races, locks, and semaphores |
| 3 | 17 January 2018 | Models of concurrency & synchronization algorithms |
| 4 | 22 January 2018 | Synchronization problems with semaphores |
| 5 | 24 January 2018 | Monitors |
| 6 | 29 January 2018 | Introduction to functional programming in Erlang |
| 7 | 31 January 2018 | Message-passing concurrency in Erlang |
| 8 | 5 February 2018 | Synchronization problems with message passing |
| 9 | 5 February 2018 | Parallelizing computations |
| 10 | 12 February 2018 | Parallel linked lists |
| 11 | 14 February 2018 | Lock free programming |
| | 19 February 2018 | Guest lecture: Niklas Gustavsson, Spotify |
| 12 | 21 February 2018 | Verification of concurrent programs |
| 13 | 26 February 2018 | Models and languages of concurrent computation |

Labs

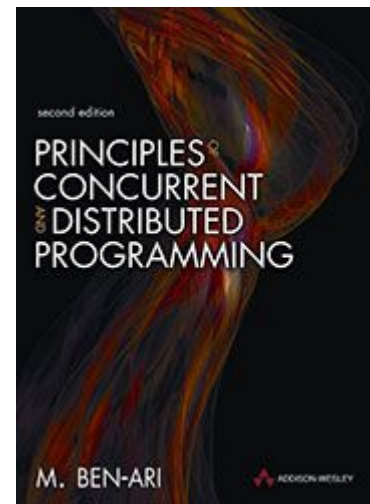
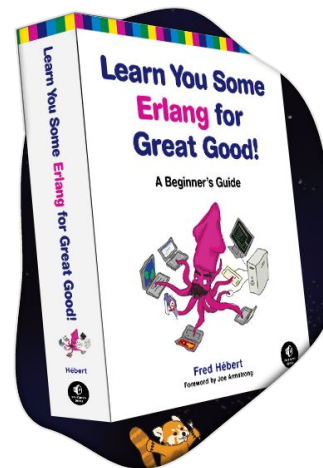
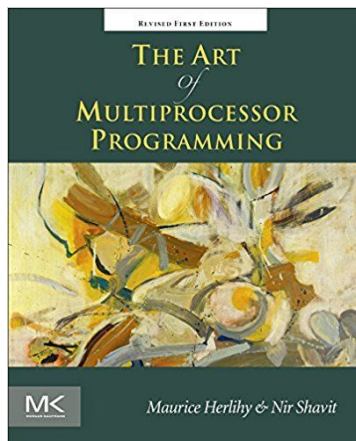
1. Trainspotting (Java)
2. CCHAT (Erlang)
3. A-mazed (Java)

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

- Register your group (2 persons) using Fire (see Fire URL in course homepage)
- Make sure to check the TA supervision schedule

Notes and reading material

- Slides of each lecture: on the [website](#)
- Ben-Ari: *Principles of concurrent and distributed programming*, 2nd edition
- Hébert: *Learn you some Erlang for great good* (freely available online)
- Herlihy & Shavit: *The art of multiprocessor programming*



Exam

- Open-book (2 books maximum), some notes allowed (4 sheets of paper maximum)
- All topics in the lectures can be examined (except the guest lecture)
- See exams of previous years for examples

- Exam dates: see TimeEdit
- Exam grading: see course [website](#)

Computing resources

- Install recent versions of Java and Erlang/OTP on your computers
- Try out the examples presented in class (some complete examples will be available on the website together with each lecture)
- Lab 1 (Trainspotting) requires a simulator, which runs in the laboratory computers
- The course website helps you set this up to work on your own computer as well