Practical Information

Lecture 0 of TDA384/DIT391

Principles of Concurrent Programming

Nir Piterman and **Gerardo Schneider**

Chalmers University of Technology | University of Gothenburg

SP3

Canvas Room and Course Website

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

Canvas	Announcements, discussion forum, videos
CTH login	https://chalmers.instructure.com/courses/27957
GU login	https://chalmers.instructure.com/courses/12523~27957
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/27957/discussion_topics

https://canvas.gu.se/courses/12523~27957/discussion_topics

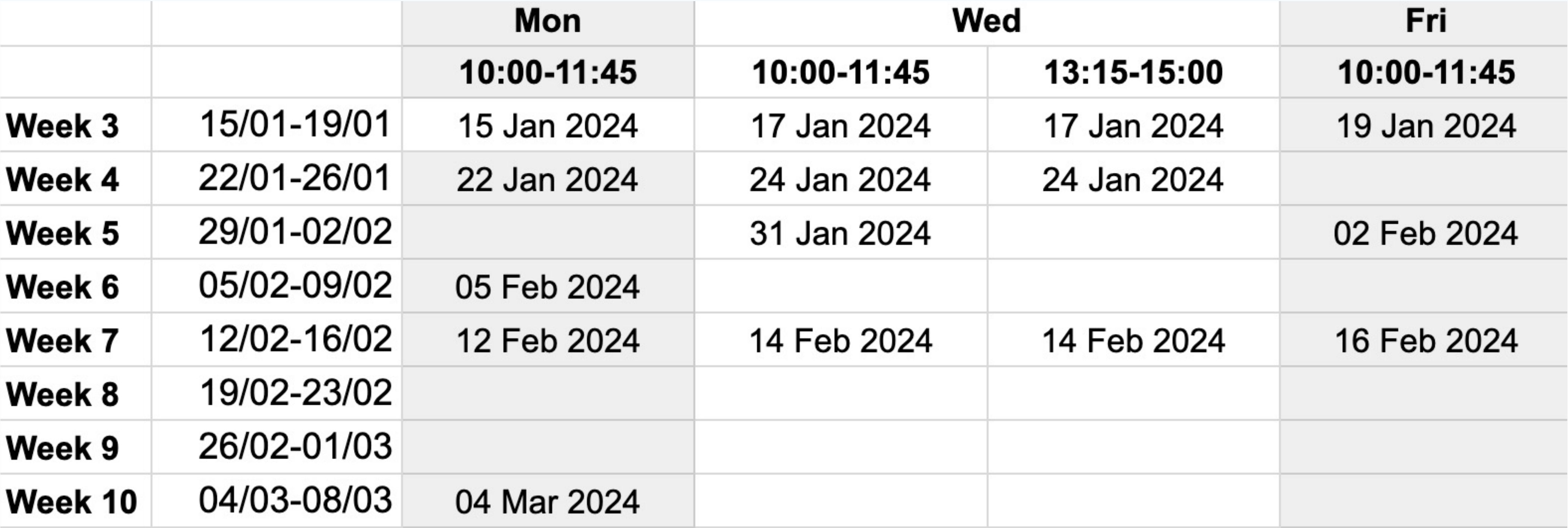
The forum URL is of course linked from the course website.

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 **HB3**
- One lecture in **HA1**
- One lecture in **HC4**
- Check out **TimeEdit**!

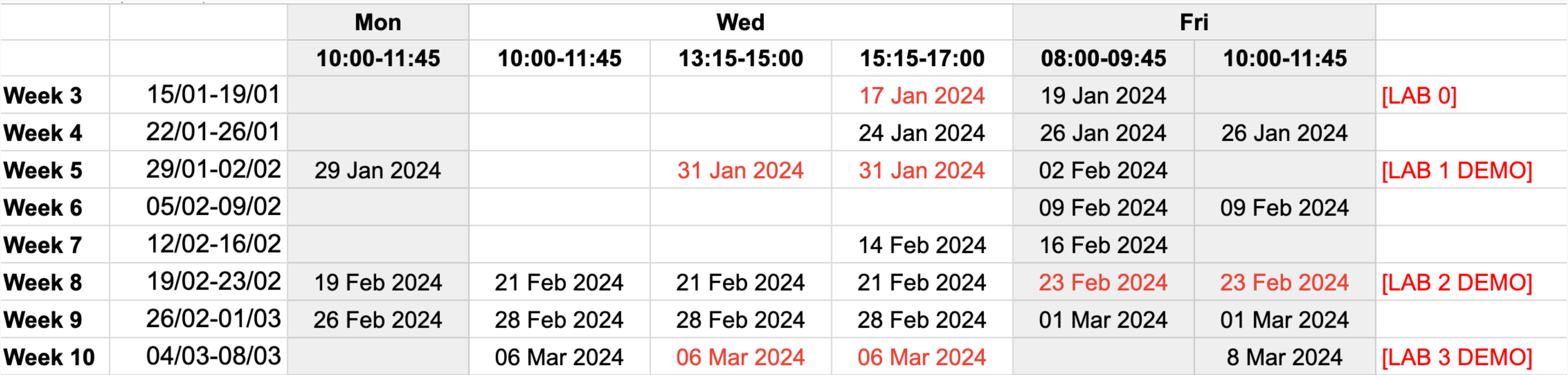


		Mon	Wed		Fri
		10:00-11:45	**10:00-11:45**	**13:15-15:00**	**10:00-11:45**
Week 3	15/01-19/01	15 Jan 2024	17 Jan 2024	17 Jan 2024	19 Jan 2024
Week 4	22/01-26/01	22 Jan 2024	24 Jan 2024	24 Jan 2024	
Week 5	29/01-02/02		31 Jan 2024		02 Feb 2024
Week 6	05/02-09/02	05 Feb 2024			
Week 7	12/02-16/02	12 Feb 2024	14 Feb 2024	14 Feb 2024	16 Feb 2024
Week 8	19/02-23/02				
Week 9	26/02-01/03				
Week 10	04/03-08/03	04 Mar 2024			

Labs

- At the beginning of the course, register in Fire
- Lab assistance requests –
  - Create a Zoom meeting without 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 (without 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



		Mon	Wed			Fri		
		10:00-11:45	10:00-11:45	13:15-15:00	15:15-17:00	08:00-09:45	10:00-11:45	
Week 3	15/01-19/01				17 Jan 2024	19 Jan 2024		[LAB 0]
Week 4	22/01-26/01				24 Jan 2024	26 Jan 2024	26 Jan 2024	
Week 5	29/01-02/02	29 Jan 2024		31 Jan 2024	31 Jan 2024	02 Feb 2024		[LAB 1 DEMO]
Week 6	05/02-09/02					09 Feb 2024	09 Feb 2024	
Week 7	12/02-16/02				14 Feb 2024	16 Feb 2024		
Week 8	19/02-23/02	19 Feb 2024	21 Feb 2024	21 Feb 2024	21 Feb 2024	23 Feb 2024	23 Feb 2024	[LAB 2 DEMO]
Week 9	26/02-01/03	26 Feb 2024	28 Feb 2024	28 Feb 2024	28 Feb 2024	01 Mar 2024	01 Mar 2024	
Week 10	04/03-08/03		06 Mar 2024	06 Mar 2024	06 Mar 2024		8 Mar 2024	[LAB 3 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)

David Lidell

Alejandro Luque

Guilherme da Silva

Reza Rezvan

Ali Saaeddin

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

- TKDAT: August Ådahl (august.adahl@outlook.com)
- TKDAT: Isac Åkesson Jansen (isacakessonjansen@gmail.com)
- …

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

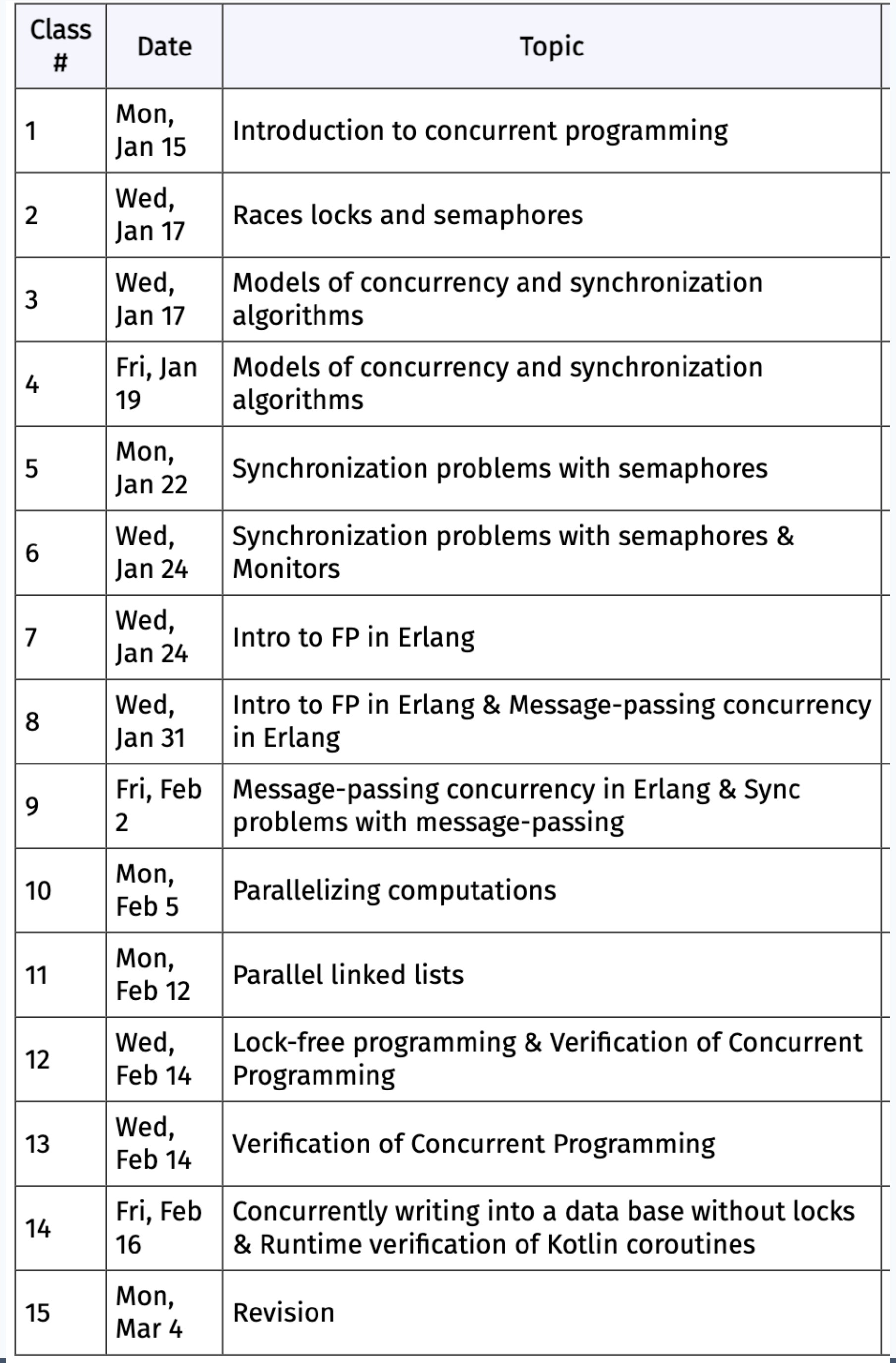
- Introduction to concurrency

- 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

Lectures



Class #	Date	Topic
1	Mon, Jan 15	Introduction to concurrent programming
2	Wed, Jan 17	Races locks and semaphores
3	Wed, Jan 17	Models of concurrency and synchronization algorithms
4	Fri, Jan 19	Models of concurrency and synchronization algorithms
5	Mon, Jan 22	Synchronization problems with semaphores
6	Wed, Jan 24	Synchronization problems with semaphores & Monitors
7	Wed, Jan 24	Intro to FP in Erlang
8	Wed, Jan 31	Intro to FP in Erlang & Message-passing concurrency in Erlang
9	Fri, Feb 2	Message-passing concurrency in Erlang & Sync problems with message-passing
10	Mon, Feb 5	Parallelizing computations
11	Mon, Feb 12	Parallel linked lists
12	Wed, Feb 14	Lock-free programming & Verification of Concurrent Programming
13	Wed, Feb 14	Verification of Concurrent Programming
14	Fri, Feb 16	Concurrently writing into a data base without locks & Runtime verification of Kotlin coroutines
15	Mon, Mar 4	Revision

- 13 lectures + 1 revision lecture + 1 guest lecture (Lecture 14 - TBD)
- 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 17 at 15:15**

2. Erlang tutorial: **Wed Jan 24 at 15:15**

NOTE: THE TUTORIALS WILL BE GIVEN IN ONE OF THE ROOM, BUT SINCE THERE ARE 3 ALLOCATED ROOMS, IT WILL BE DONE VIA ZOOM

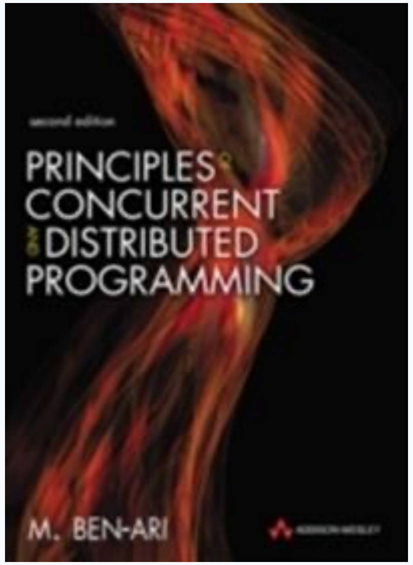
-> PLEASE BRING EARPHONES!

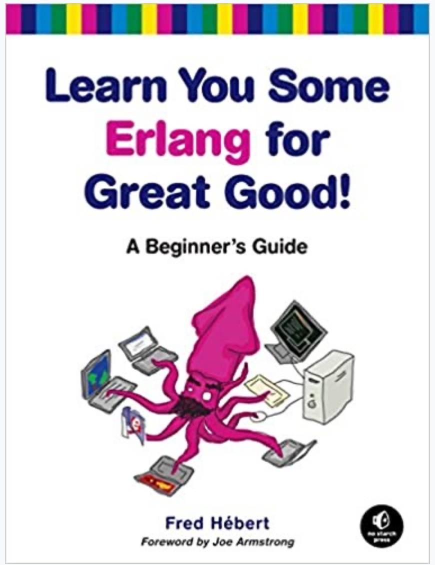
Slides and Reading Material

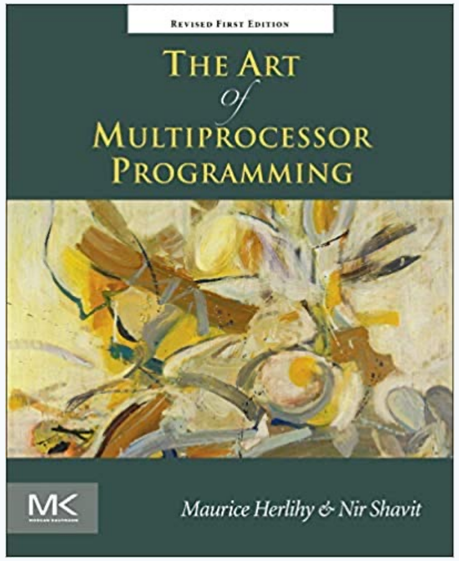
Lecture slides: will be on the website

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




Learn You Some Erlang for Great Good!
A Beginner's Guide
Fred Hébert
Foreword by Joe Armstrong


Revised First Edition
The Art of Multiprocessor Programming
MK
Maurice Herlihy & Nir Shavit

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:
  - **11 March 2024**
  - **22 August 2024** (re-exam)
  - **October 2024** (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

Check the webpage regularly!

- And of course, the Java Tutorial

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