Learning outcomes

The aim of the course:

- Functional programming:
  - recursion, pattern matching, functions as first class objects, closures, higher order functions and immutable data structures
  - implement selected algorithms in a functional programming language
- Concurrent programming:
  - advantages and disadvantages
  - message passing, actors model
  - design, implement, test and debug

Why do we need a course in functional and concurrent programming?
why concurrency

Two reasons:

A tool to model interactive services.

Hardware can utilize concurrency to speed-up computations.

To build a good game engine, you need to master concurrency

Literature

This course is not about a particular language but ....

... we need a language to experiment with.

- Introducing Elixir - Getting Started in Functional Programming
- Simon St. Laurent, J. Eisenberg
- O’Reilly Media

Best way to learn the principles of biking is riding a bike.

Lectures

We will have 14 lectures that will cover the following aspects:

- Functional programming
- Programming techniques
- Complexity
- Concurrency
- Parallelism
- Servers

Seminars

To take part in the seminar sessions, you should:

- solve an assigment before the seminar

The seminars are not compulsory but, if you attend you should have prepared well.

Collaborate in the implementation.
Seminars

- Huffman coding: implementing a compression algorithm
- A meta-interpreter: Elixir interpreter in Elixir
- Dining philosophers: introduction to concurrency
- A Mandelbrot image: parallelism
- A small web server: easier than you think

Exam

A closed book exam – where writing code by hand will be a large part.

Final grade is based on written exam.

and finally

Two student representatives for course board. We will meet a couple of times during the course so that you can give feedback.

Don’t use KTH Social for feedback or questions - use Canvas.