Advanced Functional Programming

Period 2, 5 hp (33% pace)

Instructors
Lars-Henrik Eriksson
(lhe@it.uu.se)

Advanced func. prog. techniques

- Monads and continuations
  - monads are a way of "simulating" side effects
  - continuations are a way of achieving the effect of exceptions or "go to"
- Lazy evaluation and infinite data structures
  - Nothing is computed until it is "needed". Potentially infinite data structures are constructed on the fly as they are used.
- Parallell processes
  - Functional programming is particularly suitable for programming with independent processes.
- Metaprogramming
  - Programs interpret, modify or analyse themselves or other programs

Languages used

- Lisp (dialect)
  - The classic functional language with origins in the 1950's.

- Haskell
  - The current "mainstream" state of the art functional language

- Erlang
  - A parallel functional language designed for use in high-availability applications such as telephone exchanges.

Course setup

- Flipped classroom
  - students read up on written material and/or
  - listen to recorded lectures…
  - …in advance of the tutorial sessions

- Tutorial sessions
  - the teacher gives examples and explanations
  - students practice

- Programming assignments in each of three different languages

Examination

- the programming assignments

- an essay

- several oral presentations

Why should you take this course?

- Gain a deeper understanding of
  - functional programming languages
  - advanced programming techniques
- For further studies in programming languages or theoretical computer science.
- It's fun!

Prerequisites

- A previous course in functional programming
  - E.g. Functional Programming I in period 1.