

1DL450 Advanced Functional Programming

More "Fun"ctional Programming
with Erlang, Racket, and Haskell

Administration Matters

- Lecturers:
 - Kostis Sagonas (kostis@it.uu.se)
- Permanent link to course home page:
<http://www.it.uu.se/edu/course/homepage/avfunpro/>
- Assistant:
 - Albert Mingkun Yang (albert.yang@it.uu.se)
 - responsible for the labs and the assignments
 - send to him questions about assignments

Course Structure

- We will examine three functional languages:
 - **Erlang**: concurrent, pragmatic, made-in-Sweden
 - **Racket**: a new offspring of the Lisp/Scheme family
 - **Haskell**: a nicely designed, lazy, pure FP language
- We will try to focus on the most interesting aspects of these languages:
 - **Erlang**: concurrency, scalability, reliability, testing technology and tools
 - **Racket**: macros, DSLs, continuations, contracts
 - **Haskell**: type system, monads, laziness, parallelism
- ~ 4 lectures and 1 lab for each language

Grading Scheme

- Three assignments ($3 \times 20 = 60\%$ of the grade)
 - Done individually
 - Simple programming tasks & tasks from the material covered in the lectures
 - Electronic hand-in to the student portal before the corresponding deadline
 - 10 "free" late days!
- Final "larger" project (40% of the grade)
 - Done in pairs, if you wish
 - In your choice of two of the three languages
 - Deadline: **early/mid January 2018** (TBA)
 - Brief oral presentation

Grading Scheme

- In the assignments part ($3 \times 20 = 60$ pts max)
 - 60 - 51 gets a 5
 - 50 - 41 gets a 4
 - 40 - 30 gets a 3
 - 29 - 0 gets a U (fail)
- In the project part (40pts) which can be done in pairs
 - 40 - 20 gets a G (pass)
 - 19 - 0 gets a U (fail)
- Final grade: based on the sum of the two parts
 - 100 - 81 gets a 5
 - 80 - 66 gets a 4
 - 65 - 50 gets a 3
 - 49 - 0 gets a U (fail)

Academic Honesty

- For the assignments you will work alone
- For the project you are allowed to work in pairs (but no threesomes/foursomes/...)
- Don't use work from uncited sources
 - Including old assignments or from the web



Why Study Advanced Functional Programming?

- Increase your knowledge of features and programming techniques that modern functional languages offer and the things they can do
- Learn three new languages and think about their similarities and differences
- Have fun!