

## Examination by programming assignments

Previously this course has been very traditional with respect to its theoretical contents with lectures, problem solving seminars, and an exam. This year I will try to offer a more “programming assignment approach” for those interested in implementation.

The official course syllabus and requirements cannot be changed on short notice so there will be lectures, seminars, and an exam similar to previous years, and all programming assignments will be voluntary.

For those who complete the programming assignments there will be an alternative exam. The main purpose of this exam will be to check that you have done the assignments and understood what you have done. I will expect everybody to pass this exam.

There will be some optional parts in the assignment that are required for a higher grade than 3 (or G). There will be a deadline for each assignment. A deadline is negotiable until it has passed.

You may use any program language for the implementations, but you are supposed to manage on your own. I believe that a functional programming language such as Haskell is best suited for the purpose, but any high level language with good data abstraction facilities should be fine.

I plan to publish six or seven assignments on a weekly basis on the following topics.

1. Denotational semantics for expressions. This is about modeling. If you have taken a course on object oriented modeling (eda061/edaf10) this should be familiar stuff.
2. Operational semantics.
3. Denotational semantics,
4. Functional programming semantics. This is about an execution model for functional languages like Lisp.
5. Logic programming semantics. This is about an execution model for logical languages like Prolog.
6. ...

I will not do the assignments myself so there might be problems that are too difficult to handle. If you believe that you have encountered one of these, please contact me.