



ID2202 Compilers and Execution Environments 7.5 credits

Kompilatorer och exekveringsmiljöer

This is a translation of the Swedish, legally binding, course syllabus.

Establishment

On 04/21/2020, the Head of the EECS School has decided to establish this official course syllabus to apply from autumn semester 2020, registration number: J-2020-0568.

Grading scale

A, B, C, D, E, FX, F

Education cycle

Second cycle

Main field of study

Computer Science and Engineering

Specific prerequisites

- Completed course in basic computer engineering/architecture equivalent to IS1200 Computer Hardware Engineering.
- Completed course in algorithms and data structures corresponding to ID1020.

Active participation in a course offering where the final examination is not yet reported in LADOK is considered equivalent to completion of the course. This applies only to students who are first-time registered for the prerequisite course offering or have both that and the applied-for course offering in their individual study plan.

Language of instruction

The language of instruction is specified in the course offering information in the course catalogue.

Intended learning outcomes

After passing the course, the student should be able to

- use methods for lexical, syntactic and semantic analysis
- use methods for generation of machine code
- use methods for optimizing programs
- give an account of common components in execution environments

in order to

- obtain an understanding of how a programming language is implemented as well as for the general theories that are used and how these can be applied.

For higher grades, the student should design more complex components of a compiler. The details are specified in the course memo.

Course contents

The course covers technologies for implementation of programming languages by means of compilers, both for real and virtual execution environments, technologies to read, understand, translate, improve as well as execute programs:

- To read programs: lexical analysis and syntax analysis. Finite state machines, regular expression context free grammars, LL and LR-parsing.
- To understand programs: semantic analysis, type checking.
- To translate programs: machines and instructions.
- Intermediary code, choice of instructions, conventions for procedure calls.
- To improve programs: machine independent optimisations; computer-oriented optimisations (register allocation, scheduling of instructions).
- To execute programs: virtual execution environments and runtime systems. Memory management, garbage collection, to load and link programs, just-in-time compilation.

Examination

- DAT1 - Computer based exam, 7.5 credits, grading scale: A, B, C, D, E, FX, F

Based on recommendation from KTH's coordinator for disabilities, the examiner will decide how to adapt an examination for students with documented disability.

The examiner may apply another examination format when re-examining individual students.

If the course is discontinued, students may request to be examined during the following two academic years.

Ethical approach

- All members of a group are responsible for the group's work.
- In any assessment, every student shall honestly disclose any help received and sources used.
- In an oral assessment, every student shall be able to present and answer questions about the entire assignment and solution.