



*Course syllabus*

# Optimerande kompilatorer Optimising Compilers

**EDAN75, 7,5 credits, A (Second Cycle)**

**Valid for:** 2023/24

**Faculty:** Faculty of Engineering, LTH

**Decided by:** PLED C/D

**Date of Decision:** 2023-04-18

## General Information

**Language of instruction:** The course will be given in Swedish

## Aim

The purpose of the course is that the students should understand the possibilities and limitations of modern optimising compilers, and thereby understand what they must optimise manually and what the compiler can do automatically.

## Learning outcomes

### *Knowledge and understanding*

For a passing grade the student must

- understand which algorithms and data structures are used when modern optimising compilers are implemented, and
- understand which language constructs limit the compiler's ability to optimise code.

### *Competences and skills*

For a passing grade the student must

- be able to analyse benchmark programs in order to understand which optimisation algorithms are suitable to implement, and
- be able to implement optimisation algorithms on SSA form, validate that they are implemented correctly and to measure their performance effects.

### *Judgement and approach*

For a passing grade the student must

- without assistance be able to read and understand a scientific article on optimising compilers and to present it to the other students,
- be able to make a rough estimation of how long it may take to implement the ideas of the article, and make a judgement of whether that amount of work is worthwhile,
- and finally to determine whether the optimisation ideas can be suitable to perform by hand, if the compiler cannot do it automatically.

## Contents

Control-flow analysis, data-flow analysis, dependence analysis, alias analysis, redundancy elimination, loop optimisations, procedure optimisations, register allocation, code scheduling, optimisation for object oriented languages, optimisation of the memory hierarchy and vectorisation for processors with SIMD instructions.

## Examination details

**Grading scale:** TH - (U,3,4,5) - (Fail, Three, Four, Five)

**Assessment:** To pass the course it is required to pass the compulsory course items (exercises, laboratory work, project) and the oral exam. The project must be completed within one month after the end of the course. The final grade of the course is based on the result of the oral exam and a additional voluntarily project.

The examiner, in consultation with Disability Support Services, may deviate from the regular form of examination in order to provide a permanently disabled student with a form of examination equivalent to that of a student without a disability.

### Parts

**Code:** 0122. **Name:** Examination.

**Credits:** 4,5. **Grading scale:** TH. **Assessment:** Approved examination. **Contents:** Oral examination.

**Code:** 0222. **Name:** Compulsory Course Items.

**Credits:** 3. **Grading scale:** UG. **Assessment:** Approved compulsory itemns. **Contents:** Exercises, laboratory work and project.

## Admission

### Admission requirements:

- EDAF05 Algorithms, data structures and complexity or EDA027 Algorithms and Data Structures

**The number of participants is limited to:** No

**The course overlaps following course/s:** EDA230

## Reading list

- Jonas Skeppstedt: An Introduction to the Theory of Optimizing Compilers, with performance measurements on POWER. Skeppberg, 2020, ISBN: 978-1537091129.

## Contact and other information

**Course coordinator:** Jonas Skeppstedt, [jonas.skeppstedt@cs.lth.se](mailto:jonas.skeppstedt@cs.lth.se)

**Course homepage:** <http://cs.lth.se/edan75>

**Further information:** Compulsory course items: Exercises, labs and a project.