DD2458 Problem Solving and Programming under Pressure 9.0 credits

Problemlösning och programmering under press

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

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

Establishment

On 2019-10-15, the Head of School of EECS has decided to establish this official course syllabus to apply from the autumn semester 2020 (registration number J-2019-2097).

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 algorithms and complexity equivalent to DD2350/DD2352.

Language of instruction

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

Intended learning outcomes

The general aim of the course is that the students both should be able to use programmering as a tool for problem solving and should be able to apply theoretical knowledge from other computer science courses to solve practical problems. The course has a large focus on going the whole path from theory (in the form of algorithm design) to practice (in the form of a working program).

After passing the course, the student should be able to

- use algorithm design methods such as greedy algorithms, dynamic programming, divide and conquer, and combinatorial search to design algorithms in order to solve given problems,
- use basic algorithms in fields such as graph theory, number theory and geometry on given problems and adapt them to problem-specific circumstances,
- analyse the efficiency of different algorithms to decide which ones are sufficiently efficient in a given context,
- compare different problems with respect to difficulty,
- implement algorithms and data structures given abstract specifications,
- identify bugs in others' solution attempts on problems,
- communicate with others during problem solving in groups,
- present algorithms, data structures and problems orally in a concise and lucid way.

The intended learning outcomes are achieved by solving a large number of assignments during the course, implementing a small library of algorithms, solving problems in small groups during "problem sessions", and presenting solutions to homework assignments.

Course contents

Algorithms: computational geometry, graph algorithms, number theoretic algorithms, string matching. Design and analysis of algorithms: dynamic programming, amortised analysis, judging reasonableness. Programming skills, mainly in C++ and Java.

Examination

- LAB2 - Lab assignments and problem solving sessions, 4.5 credits, grading scale: A, B, C, D, E, FX, F
- ÖVN1 - Exercises, 4.5 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.

Examination can only take place in connection with an ongoing course offering.

Transitional regulations

LAB2 corresponds to the earlier component LAB1 and may replace this.

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.