



DD1320 Applied Computer Science 6.0 credits

Tillämpad datalogi

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 2020-10-13, the Head of School of EECS has decided to establish this official course syllabus to apply from the spring semester 2021 (registration number J-2020-2220).

Grading scale

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

Education cycle

First cycle

Main field of study

Information Technology, Technology

Specific prerequisites

Completed course in programming equivalent to DD1310/DD1311/DD1312/DD1314/DD1315/DD1316/DD1318/DD1331/DD100N/ID1018.

Active participation in a course offering where the final examination is not yet reported in LADOK is considered equivalent to completion of the course.

Registering for a course is counted as active participation.

The term 'final examination' encompasses both the regular examination and the first re-examination.

Language of instruction

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

Intended learning outcomes

In this course you will broaden your knowledge about algorithms, data structures and programming. The programming language Python is used in the course.

After the course you will have learned to

- systematically test programs to discover errors,
- use abstraction as a tool to simplify programming,
- choose a suitable algorithm for a given problem,
- compare algorithms with respect to time and memory usage,
- describe different algorithms for searching and sorting and their properties,
- devise and implement recursive algorithms,
- write a small BNF syntax (less than 10 rules) for a formal language and write a program to check if an expression conforms to the syntax,
- model real problems as search problems and implement algorithms for breadth first, depth first and best first search,
- describe basic compression algorithms and know in which types of compression they are used,
- construct an automaton for text search and describe how it works,
- implement and use stacks and queues,
- implement insertion, traversal and search operations in binary search trees and general trees, and use these,
- implement and use hash tables and hash functions,
- use priority queues,
- use simple encryption algorithms, and explain the principles of asymmetric encryption
- identify problems where the above mentioned data structures are of use and construct simple algorithms with these,

so that you will:

- become proficient in solving problems with programming,
- be able to use computer science methods in projects in different applications and
- acquire sufficient grounding to be able to enroll in follow-up courses in computer science.

Course contents

Algorithms and data structures: A systematic study of the concepts abstract data types, stacks, queues, lists, trees, searching, sorting, and recursion using the knowledge the students have from the course Programming Technique. Hashing. Priority queues. Search trees. Problem trees. Text search. Simple syntax analysis. Algorithm analysis.

Programming: Program quality. Abstraction. Modularization. Testing. Documentation. Exceptions. System calls. Standard library.

Examination

- LABD - Programming assignments, 3.0 credits, grading scale: A, B, C, D, E, FX, F
- TEN1 - Examination, 3.0 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.

Under special circumstances, other examination formats may be used.

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.