DD1328 Fundamentals of Computer Science for Scientific Computing 9.0 credits

Grundläggande datalogi för tekniska beräkningar

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

Establishment

The official course syllabus is valid from the spring semester 2026 as decided by the Director of First and Second Cycle Education: HS-2025-1999, 3.2.2. Date of decision: 2025-10-14

Grading scale

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

Education cycle

First cycle

Main field of study

Technology

Specific prerequisites

Knowledge and skills in basic programming, 5 credits, equivalent to completed course DD1333/DD1310-DD1319/DD1331/DD1337/DD100N/ID1018/ID1022.

Intended learning outcomes

After passing the course, the student should be able to

- systematically test programs to discover errors
- use abstraction as a tool to simplify the programming
- select an appropriate algorithm for a given problem
- compare algorithms with regard to time and memory usage
- describe and implement different algorithms for search and sorting
- formulate and implement recursive algorithms
- write and use simple BNF syntax
- implement, and design algorithms for, basic data structures
- design and implement simple parallel programs

in order to

- become a good problem solver using programming
- be able to use computational methods in application projects
- take advanced courses in computational mathematics, machine learning and theoretical computer science.

Course contents

In this course, the students will build on their knowledge of algorithms, data structures and program design, learn basics of parallel and distributed programming and prepare for courses in computational mathematics, machine learning and theoretical computer science. Programming languages in the course are Python and Go.

Algorithms and data structures: A systematic overview of the concepts abstract data types, stacks, queues, lists, trees, searching, sorting and recursion based on the knowledge the students acquired in the course Fundamentals of programming. Hashing. Priority queues. Search trees. Problem trees. Text searching. Simple syntactical analysis. Algorithm analysis. Cryptography.

Program Design: Program quality. Abstraction. Modularisation. Testing. System calls. Standard modules.

Parallel Programming: Basic knowledge of how a computer is working and what it is made up of, both from hardware and software perspectives. Introduction to parallel and distributed programming with processes

Examination

- LABB - Programming Assignment, 2.0 credits, grading scale: A, B, C, D, E, FX, F
- LABC - Programming Assignment, 2.0 credits, grading scale: A, B, C, D, E, FX, F
- HEM3 - Take-home Assignment, 3.0 credits, grading scale: P, F
- LABA - Programming Assignment, 2.0 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.

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.