



DD2325 Applied Programming and Computer Science 7.5 credits

Tillämpad programmering och 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

The official course syllabus is valid from the autumn semester 2023 in accordance with the decision by the Head of School: J-2023-0600. Date of decision: 09/03/2023

Grading scale

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

Education cycle

Second cycle

Main field of study

Computer Science and Engineering

Specific prerequisites

- Knowledge in basic mathematics, 15 higher education credits.

- Knowledge and skills in programming, 6 higher education credits, equivalent to completed course
DD1310/DD1311/DD1312/DD1314/DD1315/DD1316/DD1318/DD1331/DD100N/ID1018.
- The upper secondary course English B/6.

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

- write structured programmes in Python
- use classes and objects in program design
- troubleshoot programs systematically
- 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 algorithms for searching and sorting
- design and explain recursive algorithms
- describe different data structures in problem-solving contexts, such as stack, queue, tree, hash table, priority queue and graph
- implement and use different data structures in problem-solving contexts such as stack, queue, tree and hash table
- model problems as search problems and implement algorithms for graph search, such as breadth-first search and depth-first search

in order to master problem solving with programming, be able to use computational methods in applied contexts and acquire sufficient prior knowledge to be able to take advanced courses in computer science.

Course contents

Programming methodology in Python, containing aspects as programme quality, testing and troubleshooting strategies. Examples of both imperative and object-oriented programming are included, as well as programming with recursion. Algorithms that handle searching and sorting related to different data structures. Examples of treated data structures are stacks, queues, trees, hash tables and graphs. Algorithms for compression. Applications in computer science and numerical analysis.

Examination

- HEM1 - Home work, 3.0 credits, grading scale: A, B, C, D, E, FX, F
- LABB - Laboratory work, 4.5 credits, grading scale: P, 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.

Transitional regulations

HEM1 replaces TEN1 for students who have not passed this module.

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.