DD1366 Programming Paradigms 6.0 credits

Programmeringsparadigm

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 2024 in accordance with head of school decision: J-2022-2265.Date of decision: 2022-09-20.

Grading scale

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

Education cycle

First cycle

Main field of study

Technology

Specific prerequisites

Knowledge and skills in programming, 6 credits, equivalent to completed course DD1337/DD1310-DD1319/DD1321/DD1331/DD100N/ID1018. Knowledge in basic computer science, 6 credits, equivalent to completed course DD1338/DD1320-DD1328/DD2325/ID1020/ID1021. Active participation in a course offering where the final examination is not yet reported in LADOK is considered equivalent to completion of the course. Being registered for a course counts 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

After passing the course, the student should be able to:

- apply and explain general concepts in programming, in particular flow of control, recursion, interpretation, paradigms and models of computation
- apply and explain basic concepts in functional programming, in particular pure functions, referential transparency, higher order functions, immutability, types
- apply and explain basic concepts in formal languages and syntax analysis, in particular automata, regular expression, grammars, lexical analysis and recursive descent
- write own client/server programs and explain basic concepts in web programming

in order to

- obtain a broader perspective on programming
- be able to assess which paradigm and which programming language that is appropriate to solve a certain assignment
- be able to use adequate programming style in a chosen programming paradigm
- be able to participate in discussions about programming paradigms, history of programming languages, language definition, properties of type systems, principles of language design, language translation, programming principles and programming concepts actively

Course contents

Functional programming: the function concept, higher order functions, currying, evaluation strategies, streams, pattern matching, overloading, polymorphism, interpretation, lazy evaluation, types and type classes. Formal languages and syntax analysis: automata, regular expressions, grammars, lexical analysis, recursive descent, classes of languages. Web programming. Language translation: interpretation, compilation and linking. Programming paradigms control flow, subprograms, recursion, classes, types, computational models and memory models

Examination

- LAB1 - Programming assignments, 3.5 credits, grading scale: A, B, C, D, E, FX, F
- MAS1 - Mastery tests, 2.5 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.

Mastery tests are individual assignments that are reported both in writing and orally

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.