DD1361 Programming Paradigms 7.5 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

Course syllabus for DD1361 valid from Autumn 2012

Grading scale

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

Education cycle

First cycle

Main field of study

Technology

Specific prerequisites

For single course students: completed upper secondary education including documented proficiency in Swedish corresponding to Swedish B, English corresponding to English A. Furthermore: 7,5 hp in mathematics and 6 hp in computer science or programming technics.

Language of instruction

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

Intended learning outcomes

After the course the student should be able to

- implement, use and explain unification, negation and sections as well as non-deterministic programming (especially within the logical paradigm),
- implement, use and explain the higher-order functions, currying, lazy evaluation, recursion, pattern matching, interpreting and type classes (especially within the functional paradigm),
- implement, use and explain memory management and compilation and linking (especially in the imperative paradigm),
- write their own client-server and use protocols and explain how they are interpreted and written,
- use and explain regular expressions and syntax analysis using recursive descent

in order to

- get a broader perspective on programming,
- judge which paradigm and which programming language that is appropriate for solving a specific task,
- use the appropriate style of programming in the selected programming paradigm,
- actively participate in discussions about programming paradigms, programming language history, language definition, properties of type systems, principles of language design, language translation, programming principles and programming concepts.

Course contents

Logic programming: unification, backtracking, negation and cuts as well as non-deterministic programming and block diagram.
Functional programming: concept of function, higher-order functions, currying, strategies of evaluation, streams, pattern matching, overloading, polymorphism, interpreting and type classes.
Imperative programming: memory management and compilation and linking.
Internet programming.
Language definition: syntax and semantics.
The principles of language design: generality, orthogonality and uniformity.
Language Translation: interpreting, compilation and linking.
Programming principles: modularity and programming style.

Course literature

To be announced at least 4 weeks before course start at the web page for the course. Last year material produced at the department was used.

Examination

- LAB2 - Laboratory Work, 4.5 credits, grading scale: A, B, C, D, E, FX, F
- TEN2 - Examination, 3.0 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.

In this course all the regulations of the code of honor at the School of Computer science and Communication apply, see: http://www.kth.se/csc/student/hederskodex/1.17237?l=en_UK.

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.