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 2008

Grading scale

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

Education cycle

First cycle

Main field of study

Technology

Specific prerequisites

The courses 2D1345/DD1345 Introduction to Computer Science and 5B1928/SF1642 Logics or the equivalent.

Language of instruction

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

Intended learning outcomes

The goals of the course are to give the students

- a systematic outline of the most important programming paradigms,
- a systematic outline of the central concepts and idioms of programming,

so that they will be able to

- a broader perspective on programming
- judge what programming paradigms and programming languages that are suitable for fulfilling a certain task,
- follow good style in the chosen programming paradigm.

Course contents

The course may be divided into four complementing blocks

- Functional programming: you will get basic skills in programming in Haskell. This includes understanding the use of higher order functions, currying, lazy evaluation, streams, pattern matching in programming constructs, and type classes.
- Logic programming: skills in Prolog programming will be derived from understanding the language' unification algorithm and you will learn to solve complex problems with simple code. This demands an insight into how negation works in Prolog and how you can use cuts and techniques such as nondeterministic programming.
- Imperative programming: This part introduce the C programming language. You will learn how to read C code and a little bit about how to write C programs. This requires that you know how to use dynamic memory without automatic garbage collection. In the computer lab you will use typical Unix tools such as 'make' and 'man' and look at how the compilation chain works.
- Internet programming: you will learn to write a basic client-server application and we introduce this paradigm with the WWW as foremost example. You will learn the differences between CGI, RMI, applets, and servlets.

In addition, we discuss general concepts of programming languages. At the end of the course, you will be able to actively participate in discussion about

- programming paradigms and history of programming languages
- syntax and semantics of programming languages
- properties of type systems
- interpretation, compilation, and linking of computer programs
- programming principles such as modularization and programming style
- programming terminology.

Course literature

To be announced at least 2 weeks before course start at the web page for the course. Previous year material produced at the department was used.

Examination

- LAB1 - Laboratory Work, 4.5 credits, grading scale: P, 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.

Other requirements for final grade

Examination (TEN1; 4,5 university credits).
Laboratory assignments (LAB1; 3 university credits).

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.