UNIVERSITY OF GOTHENBURG

COMPUTER SCIENCE AND ENGINEERING

DIT310 Models of Computation, 7.5 higher education credits

Models of Computation, 7,5 högskolepoäng

Second Cycle

Confirmation

This course syllabus is an old version, confirmed by The IT Faculty Board on 2009-09-18 and was last revised on 2012-11-12 by Department of Computer Science and Engineering. to be valid from 2013-09-02. .

Field of education: Science 100%
Department: Computer Science and Engineering

Position in the educational system

The course is a part of the Computer Science Master's programme and a single-subject course at the University of Gothenburg.
The level for the course in relation to degree requirements is Master´s degree, code A1F.
The course has course/courses at second cycle level as entry requirements.

The course can be part of the following programmes: 1) Computer Science, Master's Programme (N2COS), 2) Computer Science, Bachelor´s Programme (N1COS) and 3) No translation available (NDATM)

Main field of studies	*Specialization*
Computer Science-Algorithms and Logic	A1F, Second cycle, has second-cycle course/s as entry requirements

Entry requirements

To be eligible for the course students should have successfully completed two year studies within the subject Computer Science or equivalent.

English B level or English proficiency equivalent to IELTS 6.5 no part under 5.5 or TOEFL 575 p, TWE score 4.5 is also required.

Learning outcomes
After completion of the course the student should be able to:

1. Knowledge and understanding

- clarify the difference between an algorithm and mathematical function;
- conclude the notions of enumerable set and different properties of functions (like total, partial, surjective, injective);
- state the relationship between an inductively defined data type, primitive recursion and induction proofs;
- define the notion of computable function;
- explain Church's thesis.

2. Skills and abilities

- give the abstract syntax and computation rules for a small programming language;
- relate and implement lambda-calculus;
- implement recursive functions in lambda calculus;
- use the class of primitive recursive functions and express some simple programs in it;
- show how a simple functional language can be used as a model for computation;
- explain what a Turing machine is and how to write some simple programs for it;
- give some examples and proofs of problems which are not computable.

3. Judgement and approach

- argue for different evaluation strategies and their termination properties.

Course content
The aim of the course is to give an introduction to some basic models of computing, their syntax and semantics. Concepts like program, programming language and computing are studied from a more general and mathematical perspective. A rigorous mathematical formulation of these requires some simplification compared to practical programming languages. This is similar to Physics where you simplify the physical reality (like strings with no mass and particles without extent) in order to have a scientific base for further study. An advanced programmer needs basic knowledge of Computer Science theory in the same way as an engineer needs basic knowledge of Physics.

Form of teaching

Language of instruction: English

Assessment

The examination consists of three parts: Weekly exercise (up to 20 points), written individual assignments (up to 40 points) and an individual written exam, given in an examination hall (up to 140 points). It is necessary to have 100 points to pass the course and 170 points to pass with distinction. The points for the exercises and assingments are only valid for one year after the start of the course.

A student who has failed the same examination twice has right to request of the department a change of examiner. The request is to be in writing and submitted as soon as possible. The department is to grant such a request without undue delay.

In cases where a course has been discontinued or major changes have been made a student should be guaranteed at least three examination occasions (including the ordinary examination occasion) during a time of at least one year from the last time the course was given.

Grades

The grading scale comprises: Fail (U), Pass (G), Pass with Distinction (VG).
100 points are required to pass the course, 170 points to pass with distinction.

Regarding the application of ECTS scales, please see Vice-Chancellors decision 2007-05-28, dnr G 81976/07.

Course evaluation

The course is evaluated through meetings both during and after the course between teachers and student representatives. Further, an anonymous questionnaire is used to ensure written information. As far as possible, evaluations ar to be completed electronically. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed.

Additional information

The syllabus correspond to the syllabus of the Models of Computation course at Chalmers.