ID2010 Programming of Interactive Systems 7.5 credits

Programmering av interaktiva system

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 ID2010 valid from Autumn 2019

Grading scale

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

Education cycle

Second cycle

Main field of study

Computer Science and Engineering

Specific prerequisites

For single course students:

- Completed documented upper secondary education incl documented proficiency in English and university studies corresponding to 60 credits (hp)/2 years of study.
- Academic studies corresponding to 180 ECTS (hp) in Information Technology/Computer Science/Computer and Systems Sciences.

Language of instruction

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

Intended learning outcomes

On successful completion of this course the students should be able to:
- recognise and identify the distinguishing properties of a distributed computer system
- describe and use basic computer networking technology, with particular emphasis on TCP/IP
- extend and enhance a given networking Java application, with multiple components and asyncronous interactions
- give examples of loosely coupled ad-hoc systems, consisting of reactive and proactive components.

Course contents

The course is divided into three parts:
-distributed systems
-the agent metaphor
-programming with Java
Distributed systems can be distinguished from local (or single-host) systems by properties such as latency, memory access and partial failure and concurrency. These properties and their effects on performance and robustness are reviewed. Design issues related to division of labour, transaction semantics, central- vs distributed functionality, peer-to-peer vs client-server solutions are then treated. Approaches to heterogeneous platforms are touched upon, followed by issues related to small systems, e.g. constraints in terms of CPU, memory, power, network connectivity and GUI real estate. This is followed by a discussion of networking issues like random connectivity, wireless link layers and techniques for discovery and message exchange.
The agent metaphor as an approach to distributed system design is discussed, emphasising on the modelling and creation of systems of autonomous, reasoning, communicating and proactive entities. This raises questions about knowledge and its representations, reasoning, plans and goal-directed behaviour — in isolation and in collaboration with other entities — interspersed with periods of deactivation, hibernation or transport. Implementation issues involve the proper selection and integration of runtime components, e.g. multi-threading, implementations in multiple programming languages and state maintenance.
Two assignments are given, both involving a Java application that is to be extended or completed.

Disposition

Lectures, labs (mostly unsupervised) and seminars.

Examination

- LAB1 - Laboratory Work, 3.0 credits, grading scale: P, F
- TEN1 - Examination, 4.5 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.

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.