DD2385 Software Engineering 6.0 credits

Programutvecklingsteknik

This is a translation of the Swedish, legally binding, course syllabus.

Establishment

Course syllabus for DD2385 valid from Spring 2013

Grading scale

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

Education cycle

Second cycle

Main field of study

Computer Science and Engineering, Information Technology, Information and Communication Technology

Specific prerequisites

Single course students: 90 university credits including 45 university credits in Mathematics or Information Technology. English B, or equivalent and Swedish B, or equivalent.

Language of instruction

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

Intended learning outcomes

The overall aim of the course is to provide an understanding of basic object oriented software development including the use of design patterns, based on the Java programming language.

After the course you should be able to:

- Use object oriented principles and concepts including loose coupling, encapsulation, inheritance and polymorphism in your own programming.
- Describe and apply principles for good object oriented design.
- Describe common design patterns and choose suitable patterns for simple applications.
- Use UML class diagrams to document your own programming work in a clear and legible way.
- Interpret UML class diagrams, e.g. as introduction to new design patterns.
- Understand a variety of advanced Java features including Swing GUI components, exceptions, network programming and concurrency and be able to use these in lab work.
- Perform an object-oriented analysis of an informal text-based software requirements document and use a Noun/Verb/Relational-Phrase methodology, to construct a data dictionary.
- Translate the information contained in a data dictionary into a UML class diagram which accurately models the same information, including aggregation, inheritance and multiplicity.

Course contents

Object oriented programming in Java. Class libraries for collections, grafical components, client-server programming and threads. UML class diagrams.

Design patterns for object oriented program development and design criteria.

XML. Common object oriented methodologies. Testing. Other UML than class diagrams.

Lab work in Java. Alls labs are documented with UML.

Course literature

Course literature will be announced at least 4 weeks before course start at course web page. The year before this Brande, Software design, was used.

Examination

- TEN1 - Examination, 3.0 credits, grading scale: A, B, C, D, E, FX, F
- LABA - Laboratory Work, 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.

If the course is discontinued, students may request to be examined during the following two academic years.

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.