ML1301 Object Oriented Programming 7.5 credits

Programmering fortsättningskurs

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

Establishment

Course syllabus for ML1301 valid from Autumn 2008

Grading scale

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

Education cycle

First cycle

Main field of study

Technology

Specific prerequisites

ML1300/HI1900/6S2950, Computer Programming, Basic Course, or corresponding course.

Language of instruction

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

Intended learning outcomes

The aims of this course are to provide the participants with knowledge about object-oriented syntax and programming techniques, and also basic knowledge about object-oriented design.

To obtain the grade E, the student should be able to

- describe the different phases in object-oriented software development.
- develop simple models, that can be used for implementation of applications, using the concepts of object-orientation.
- describe the model using the Unified Modelling Language,
- implement classes that, when possible, can be reused in similar contexts.

To obtain the grade E, the student should also, using the some object-oriented language, be able to

- use an IDE to implement, debug and execute object-oriented applications.
- implement structured applications, using object-oriented principles like data abstraction and encapsulation, that are easy to maintain and further develop.
- use inheritance to extend and override the functionality of classes.
- implement polymorphic code.
- use exception-handling to handle run-time errors.
- use streams to read and write data from/to different types of sources/targets.
- implement simple concurrent applications using threads, describe problems related to concurrent programming and how to solve these problems.
- implement event driven applications with simple graphical user interfaces.
- describe, and in simple cases implement, the Model-View-Controller pattern.

To obtain higher grades, the student should also be able to

- use the methods in object-oriented analysis and design to develop models of and implement more complex applications.

Course contents

- Basic concepts in object-oriented programming: abstract data types, encapsulation, reuse.
- IDE.
- Principles of object-oriented programming: classes/objects, relations, inheritance, polymorphism.
- Object-oriented software development: analysis, design and implementation.
- Class diagrams and relations in e.g. Unified Modelling Language.
- Exception-handling.
- Streams.
- Concurrent programming.

• Graphical user interfaces and event driven programming.

Course literature

The course literature is posted on the course's homepage at least four weeks before the course starts.
The previous academic year used the following literature:
KSierra K & Bates B, Head First Java 2nd ed. O´Reilly 2005.

Examination

• ÖVN1 - Exercises, 4.5 credits, grading scale: A, B, C, D, E, FX, 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.

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

Other requirements for final grade

Passed written exam. The exam contains both theoretical and practical portions.
Passed lab assignments. The final grade is based on all parts of the examination.

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.