HI1201 Object- Oriented Programming 6.0 credits

Objektorienterad programmering

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 HI1201 valid from Autumn 2012

Grading scale

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

Education cycle

First cycle

Main field of study

Technology

Specific prerequisites

HI1200, 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 basic knowledge about object-oriented application development and in-depth knowledge about syntax and programming techniques in an object-oriented programming language.

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

- describe the different phases in object-oriented software development
- develop models, that can be used for implementation of applications, using the concepts of object-orientation.
- describe the solution using class diagrams and dynamic diagrams, following the syntax of Unified Modeling Language

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

- implement cohesive classes with well-defined interfaces
- implement inheritance and dependencies such ass association and aggregation
- implement data strucutres such as stack, queue, list and tree
- use exception-handling to handle errors
- identify and use some basic object-oriented design patterns
- implement event driven applications
- implement applications with graphical user interfaces using the Model-View-Controller design pattern.
- use an IDE to implement, debug and execute object-oriented applications

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

- use the methods in object-oriented analysis and design to develop models of more complex applications.
- design and implement flexible and reusable software solutions using object-oriented principles

Course contents

- Basic concepts in object-oriented programming: abstract data types, encapsulation, interfaces, classes, objects, dependencies, inheritance, polymorphism
- Abstract data types such as stack, queue, list and tree
- Syntax and implementation in an object-oriented language
- Object-oriented software development: analysis, design and implementation
- Unified Modelling Language
- Exception-handling
- Graphical user interfaces and event driven programming

- Introduction to object-oriented Design Patterns

Course literature

Anges senast 4 veckor innan kursstart

Examination

- LAB1 - Exercises, 3.0 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.

TEN1, 3.0 credits (ECTS) A-F. The exam contains both theoretical and practical portions. Lab assignments, LAB1, 3.0 credits (ECTS) A-F.

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.