HI1027 Object Oriented Programming 8.0 credits

Objektorienterad programmering

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

Establishment

Course syllabus for HI1027 valid from Autumn 2024

Grading scale

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

Education cycle

First cycle

Main field of study

Technology

Specific prerequisites

HI1024 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 course will give students the basics of object-oriented thinking in programming as well as good knowledge of syntax and programming techniques in an object-oriented language.

For a passing grade, students must:

1. Describe object-oriented principles such as encapsulation, abstract data types, interfaces, inheritance, high cohesion, and low coupling

2. Based on a straightforward problem description, develop an object-oriented model for implementation, with classes, members, methods, and relationships, that follow object-oriented principles

3. Describe some common object-oriented design patterns

For a passing grade, the student must also, in specific object-oriented language, be able to:

4. Implement classes, and objects, following object-oriented principles such as encapsulation, distinct interfaces, inheritance, high cohesion, and low connectivity

5. Implement common data types such as list, queue, and stack as well as some common design patterns

6. Use exceptions to signal errors and handle them at an appropriate level in the application

7. Implement multi-threaded applications and protect shared data from synchronization issues

8. Write applications with graphical interfaces and event handling

Course contents

- Principles of object-oriented programming: encapsulation, abstract data types, interfaces, high cohesion, and low coupling
- Classes and objects, relations between objects and classes, inheritance, and interface, polymorphism
- Introduction to implementation of abstract data types and object-oriented design patterns
- Generics
- Syntax and implementation in an object-oriented language
- Excptions and exception handling
- Streams for reading from, and writing to, other devices, such as files and applications
- Thread programming
- Garphical user interfaces and event handling
- Functional programming in object-oriented languages: functional interfaces, streams, lambda expressions

Examination

- TEN1 - Written examination, 3.5 credits, grading scale: A, B, C, D, E, FX, F
- LAB1 - Mandatory laboratories, 4.5 credits, grading scale: P, 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.

Written exam, TEN1, 3.5 credits (ECTS) A-F. The exam contains both theoretical and practical portions.
Lab assignments, LAB1, 4.5 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.