



Course syllabus

Faculty of Technology

Department of Computer Science and Media Technology

2DV608 Mjukvarudesign, 7,5 högskolepoäng

Software Design, 7.5 credits

Main field of study

Computer Science

Subject Group

Informatics/Computer and Systems Sciences

Level of classification

First Level

Progression

G2F

Date of Ratification

Approved 2019-06-10

Revised 2021-06-17 by Faculty of Technology. Prerequisites are revised.

The course syllabus is valid from spring semester 2022

Prerequisites

60 credits within Computer Science including

1DV506 Problem solving and programming, 7.5 credits, and 1DV607 Object-Oriented Analysis and Design using UML, 7.5 credits

or

1DV607 Object-Oriented Analysis and Design using UML, 7.5 credits and 1DV502 Object-Oriented Programming, 7.5 credits or equivalent

Objectives

Upon completion of the course, students should be able to:

- Explain and apply fundamental concepts, principles, and techniques of Requirements Engineering (elicitation, specification, and modeling)
- Explain and apply fundamental concepts, principles, and techniques of Performance Engineering (modeling, metrics, and evaluation)
- Explain and apply fundamental concepts, principles, and techniques of Software Architecting and Design (processes, modeling, metrics, and evaluation)

Content

The course motivates the need for *quality* and elaborates on Software Engineering practices for achieving it, namely *Requirement Engineering*, *Performance Engineering*, and *Software Architecting and Design*.

The course covers the following topics:

1) Requirement Engineering

- Requirements Elicitation
- Requirements Validation and Management
- Requirements Modelling

2) Performance Engineering

- Software Performance Modeling
- Software Performance Evaluation

3) Software Architecting and Design

- Design Principles
- Software Architecture and Architectural Patterns
- Re-engineering Legacy Systems

Type of Instruction

Teaching consists of lectures, seminars, and practical work.

Examination

The course is assessed with the grades A, B, C, D, E, Fx or F.

The grade A constitutes the highest grade on the scale and the remaining grades follow in descending order where the grade E is the lowest grade on the scale that will result in a pass. The grade F means that the student's performance is assessed as fail (i.e., received the grade F).

Assessment of student performance is made through 3 different assignments (2.5 credits each). Students who do not pass the regular examination will be offered retrials close to the regular examination.

To pass the course, grade E or higher is required for all assignments. The final grade is decided as the average of the assignments.

Repeat examination is offered in accordance with Local regulations for courses and examination at the first and second-cycle level at Linnaeus University.

If the university has decided that a student is entitled to special pedagogical support due to a disability, the examiner has the right to give a customised exam or to have the student conduct the exam in an alternative way.

Course Evaluation

During the implementation of the course or in close conjunction with the course, a course evaluation is to be carried out. Results and analysis of the course evaluation are to be promptly presented as feedback to the students who have completed the course. Students who participate during the next course instance receive feedback at the start of the course. The course evaluation is to be carried out anonymously.

Credit Overlap

The course cannot be included in a degree along with the following courses of which the content fully, or partly, corresponds to the content of this course: 2DV603 Software Engineering - Design, 7,5 hp

Other

Grade criteria for the A–F scale are communicated to the student through a special document. The student is to be informed about the grade criteria for the course by the start of the course at the latest.

The course is conducted in such a way that the course participants' experiences and knowledge are made visible and developed. This means, for example, that we have an inclusive approach and strive for no one to feel excluded. This can be expressed in different ways in a course, for example by using the gender-neutral example.

Required Reading and Additional Study Material

Required reading

- Ian Sommerville. *Software Engineering (10/E)*. Pearson, 2015. ISBN: 0133943038. 16 pages
- Timothy Lethbridge, Robert Laganie`re. *Object-Oriented Software Engineering*. McGraw-Hill Science, 2005. ISBN-13: 978-0077109080. 200 pages
- Chris Birchall. *Re-Engineering Legacy Software*. Manning Publications, 2016. ISBN: 978161729250. 150 pages
- Ian Sommerville and Pete Sawyer. *Requirements Engineering - A Good Practice Guide*. ISBN-13: 978-0471974444. 200 pages.
- Edward D. Lazowska, John Zahorjan, G. Scott Graham, and Kenneth C. Sevcik. *Quantitative System Performance - Computer System Analysis Using Queueing Network Models*. Prentice-Hall. ISBN:0-13-746975-6. 50 pages
- Object Management Group. *UML profile for MARTE*. ISBN: NA. 50 pages