Design Patterns

Programme course

6 credits

Designmönster

TDDB84

Valid from: 2017 Spring semester

Determined by
Board of Studies for Computer Science
and Media Technology

Date determined
2017-01-25

Main field of study
Information Technology, Computer Science and Engineering, Computer Science

Course level
Second cycle

Advancement level
A1X

Course offered for
- Computer Science, Master's Programme
- Computer Engineering, B Sc in Engineering
- Computer Science and Engineering, M Sc in Engineering
- Information Technology, M Sc in Engineering
- Media Technology and Engineering, M Sc in Engineering
- Computer Science and Software Engineering, M Sc in Engineering
- Industrial Engineering and Management - International, M Sc in Engineering
- Industrial Engineering and Management, M Sc in Engineering
- Computer Science, Master's programme

Entry requirements
Note: Admission requirements for non-programme students usually also include admission requirements for the programme and threshold requirements for progression within the programme, or corresponding.

Prerequisites
The student must be well acquainted with programming in at least one object-oriented programming language and must have used advanced algorithms and data structures. The student must be acquainted with a modelling language (UML) for object-oriented systems. The student should understand the overall structure and organization of large programs. The student should know how to read research papers on software engineering and search information independently on new topics.

Intended learning outcomes

After the course, students shall be able to:

- identify and explain design principles and patterns in software
- apply design principles and patterns when working with software
- describe the purpose with and consequences of design patterns
- critically evaluate consequences of design patterns with respect to desired software qualities,
- analyze the relationship between design principles, design patterns, and programming language design, application frameworks, or application domains.

Course content

Software design. Design Patterns. History of design patterns. Refactoring. Testability, domain-specific languages, dependency injection, SOLID

Teaching and working methods

Students will review online lecture material and study the course literature in groups, and attend group seminars where course topics will be discussed. Students will perform a series of lab assignments in groups to analyze software components with respect to design principles and patterns, and study how design principles affect the ability to modify and extend software components.

Examination

LAB1	Laboratory Works	3 credits	U, 3, 4, 5
UPG1	Written report	3 credits	U, 3, 4, 5

Grades

Alternative-grade scale, LiU, U, 3, 4, 5

Other information

Supplementary courses

Component-based software, enterprise systems

Department

Institutionen för datavetenskap

Director of Studies or equivalent
Ahmed Rezine

Examiner
Ola Leifler

Course website and other links
http://www.ida.liu.se/~TDDB84

Education components
Preliminary scheduled hours: 40 h
Recommended self-study hours: 120 h

Course literature

Additional literature

Books

Freeman & Freeman, *Head First Design Patterns*
Gamma, Helm, et.al., (1995) *Design Patterns: Elements of Object-Oriented Software* Addison Wesley

Other

Selected articles on software design and design patterns.

Common rules

Regulations (apply to LiU in its entirety)

The university is a government agency whose operations are regulated by legislation and ordinances, which include the Higher Education Act and the Higher Education Ordinance. In addition to legislation and ordinances, operations are subject to several policy documents. The Linköping University rule book collects currently valid decisions of a regulatory nature taken by the university board, the vice-chancellor and faculty/department boards.

LiU's rule book for education at first-cycle and second-cycle levels is available at http://styrdokument.liu.se/Regelsamling/Innehall/Utbildning_pa_grund-_och_avancerad_niva.