LUNDS UNIVERSITET
Lunds Tekniska Högskola

Course syllabus

Programmeringsteknik - fördjupningskurs
Programming - Second Course

EDAA01, 7,5 credits, G1 (First Cycle)

Valid for: 2023/24
Faculty: Faculty of Engineering, LTH
Decided by: PLED C/D
Date of Decision: 2023-02-13

General Information

Main field: Technology.
Compulsory for: C1, D1, E2, Pi2
Elective Compulsory for: I3, L3, M3
Elective for: B4, BME4-sbh, F2, K4, MD4, N2, W4, BR4
Language of instruction: The course will be given in Swedish

Aim

The purpose of the course is to give the students extended knowledge of object-oriented programming techniques and of a selection of algorithms and data structures suitable for solving commonly occurring problems. The course should also provide the skills required to understand and use modern libraries for object-oriented programming.

Learning outcomes

Knowledge and understanding
For a passing grade the student must

- be able to define and explain important concepts and language constructs within object-oriented programming.
- be able to describe abstract data types such as lists, stacks, queues, priority queues, sets and maps and their applications.
- be able to describe data structures such as linked lists, binary trees, heaps and hash tables and explain how they can be used to implement fundamental abstract data types.

Competences and skills
For a passing grade the student must

- be able to apply advanced object-oriented concepts and language constructs in Java.
- be able to use modern object-oriented standard libraries.
- be able to identify appropriate abstract data typess, algorithms and data structures for solving a given problem.
- be able to implement fundamental abstract data types and algorithms in Java.
- be able to formulate and implement recursive algorithms.
- be able to write programs with a simple graphical user interface in Java.
- be able to perform simple tests using a test tool.
- be able to apply simple techniques for analysing the time complexity of algorithms and estimate their execution time.

Judgement and approach
For a passing grade the student must

- be able to construct programs which are easy to understand and to modify
- be able to evaluate proposed solutions and data representations for given problems with respect to usability and efficiency

Contents

Object oriented concepts and language constructs such as inheritance, interface, nested classes, exceptions, generics, lambda expressions and streams. Important general interfaces such as Iterable, Comparable and Comparator. Common abstract data types such as lists, stacks, FIFO queues, priority queues, sets, and map sand their corresponding interfaces and classes in the Java API. Data structures such as linked lists, binary trees, heaps, and hash tables, and how they can be used to implement fundamental abstract data types. Recursion as a technique to construct and implement algorithms, techniques such as dynamic programming and backtracking. Orientation on frameworks for graphical user interfaces and event-driven programming. Introduction to using tools for testing. Introduction to techniques for analysing the time complexity of algorithms.

Examination details

Grading scale: TH - (U,3,4,5) - (Fail, Three, Four, Five)
Assessment: For final grades, approved compulsory laboratory work, approved assignments and an approved written exam are required. The final grade for the course is based on the result of the written examination.

The examiner, in consultation with Disability Support Services, may deviate from the regular form of examination in order to provide a permanently disabled student with a form of examination equivalent to that of a student without a disability.

Parts
Code: 0108. **Name:** Compulsory Course Items.
Credits: 3. **Grading scale:** UG. **Assessment:** Approved compulsory items. **Contents:** Laboratory work and project.
Code: 0208. **Name:** Written Examination.
Credits: 4,5. **Grading scale:** TH. **Assessment:** Approved examination. **Contents:** Written examination.

Admission

Admission requirements:

- EDA011 Programming, First Course or EDA016 Programming, First Course or EDA017 Programming, First Course or EDA501 Programming, First Course or EDAA10 Computer Programming in Java or EDAA20 Programming and Databases or EDAA45 Introduction to Programming or EDAA50 Programming, First Course or EDAA55 Programming, First Course or EDAA65 Programming, First Course or EDAA70 Introduction to Programming Using Python

The number of participants is limited to: No
The course overlaps following course/s: EDA020, EDA025, EDA026, EDA027, EDA035, EDA510, EDA690, EDAA30, TFRD49

Reading list

- Provided by the department.

Contact and other information

Course coordinator: Univ.adj Anna Axelsson, Anna.Axelsson@cs.lth.se
Course homepage: http://cs.lth.se/edaa01