Course syllabus

Programmeringsteknik - fördjupningskurs Programming - Second Course

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

Valid for: 2020/21
Decided by: PLED C/D
Date of Decision: 2020-02-24

General Information

Main field: Technology.
Compulsory for: C2, D1, E2, Pi2
Elective Compulsory for: I3, L3, M3
Elective for: BME4-sbh, F2, MD4, N2, W4
Language of instruction: The course will be given in Swedish

Aim

Constructing program components which are easy to change, reusable and resource-aware requires good knowledge of advanced language constructs and programming techniques as well as data abstraction, data structures and algorithms. The aim of the course is to give students increased knowledge of object oriented programming techniques and a selection of fundamental algorithms and data structures and to give them the ability to understand and use APIs for classical data structures and algorithms in modern object oriented program libraries.

Learning outcomes

Knowledge and understanding
For a passing grade the student must

- be able to define and explain important object oriented concepts and techniques in addition to what has been covered in the first programming course
- be able to describe common abstract data types and their applications
- be familiar with existing library classes for common abstract data structures and algorithms

Competences and skills
For a passing grade the student must

- be able to apply advanced object oriented concepts and techniques and advanced language constructs in Java in her/his own programming
- be able to use frameworks for graphical user interfaces in Java
- be able to use modern object oriented standard libraries
- be able to identify appropriate algorithms and data types for solving a given problem
- be able to implement fundamental abstract data types and algorithms in an object oriented language
- be able to formulate and implement recursive algorithms
- be able to apply simple techniques for analysing the time complexity of algorithms

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 interface, inner classes, exceptions and generics. Important general interfaces such as Iterator, ListIterator, Comparable and Comparator. Overview of graphical interfaces. Recursion as a technique to construct and implement algorithms. Common abstract data types: Set, Queue, Stack, List and Dictionary and their corresponding classes in the Java API. Data structures which can be used to implement common abstract data types as arrays, linked lists, trees and hash tables. Simple algorithms for efficient sorting. Introduction to techniques for analysing the time complexity of algorithms.

Examination details

Grading scale: TH - (U,3,4,5) - (Fail, Three, Four, Five)
Assessment: Written examination. The final grade of the course is based on the result of the written examination. Compulsory course items: Laboratory exercises and assígnments.

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:** For a final grade the laboratory exercises and the assignments must be completed.
Code: 0208. **Name:** Written Examination.
Credits: 4,5. **Grading scale:** TH. **Assessment:** The final grade of the course is based on the result of the written examination. **Contents:** Written examination.

Admission

Admission requirements:

- Completed compulsory course items from EDAA45 Programming, first course or a passing grade from the written examination in that course or in EDA011/EDA016/EDA017/EDAA20/EDAA50/EDAA55 or EDA501/EDAA65

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

Reading list

- Koffman E. B., Wolfgang P.: Data Structures: Abstraction and Design using Java EEd. Wiley, 2016, ISBN: 978-1-119-18652-6. Recommended, but not mandatory.

Contact and other information

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