UNIVERSITY OF GOTHENBURG

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DIT181 Data Structures and Algorithms, 7.5 credits

Datastrukturer och Algoritmer, 7,5 högskolepoäng

First Cycle

Confirmation

This course syllabus was confirmed by Department of Computer Science and Engineering on 2017-01-05 and was last revised on 2019-02-08 to be valid from 2020-01-20, spring semester of 2020.

Field of education: Science 100%
Department: Department of Computer Science and Engineering

Position in the educational system

The course is a compulsory course in the N1SOF Software Engineering and Management Bachelor's Programme. The course is also a single subject course at the University of Gothenburg.

The course can be part of the following programmes: 1) Mathematical Sciences, Master's Programme (N2MAT), 2) Applied Data Science Master's Programme (N2ADS), 3) Bachelor's Programme in Mathematics (N1MAT), 4) Software Engineering and Management Bachelor's Programme (N1SOF) and 5) Software Engineering and Management, Bachelor's Programme (N1SEM)

Main field of studies	*Specialization*
Software Engineering	G1F, First cycle, has less than 60 credits in first-cycle course/s as entry requirements

Entry requirements

To be eligible for this course, students must have successfully completed 7.5 higher education credits in object oriented programming (for example DIT042 Object-Oriented Programming, 7.5 hec) and successfully completed 7.5 hec on basic mathematical concepts such as sets, functions, relations, graphs, logarithms and proof by induction (for example DIT022 Mathematical Foundations for Software Engineering, 7.5 hec).

Learning outcomes

On successful completion of the course the student will be able to:

Knowledge and understanding

- explain basic abstract data types and data structures such as arrays, stacks, queues, linked lists, trees and hash tables
- identify and describe algorithms in the engineering research literature, given the nature of a computational problem

Competence and skills

- implement abstract data types as interfaces, and data structures as classes, in an object-oriented programming language
- use a standard library of data structures and algorithms
- read, specify, and describe algorithms, at a higher level of abstraction than code
- choose appropriate data structures and algorithms of better complexity to improve the performance of inefficient programs

Judgement and approach

- analyze the efficiency of different implementations, for example sorting algorithms
- select methodically between different data structures and algorithms for different applications
- reflect on the importance of clarity, conciseness and efficiency in the design and documentation of algorithms

Course content

The course introduces the students to the role of data structures and algorithmic concepts in the detailed design and implementation of programs. The course has two general themes: (1) the role of algorithms in the design and development of programs; (2) the role of data structures in the implementation of algorithms.

These general themes are supported by the study of sub themes from within the field of computer science:

• Asymptotic efficiency and complexity notations.
• Algorithms recurrent in engineering research literature such as searching and sorting.
• Common data structures and abstract data types, such as arrays, stacks, queues, linked lists, trees and hash tables.
• Recursion.

An object-oriented programming language is used for implementing the data structures and algorithms covered.

Sub-courses

1. **Written exam** *(Tentamen),* 4.5 credits
   Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)
2. **Assignments** *(Inlämningsuppgifter),* 3 credits
   Grading scale: Pass (G) and Fail (U)

Form of teaching

The teaching consists of lectures, group work, exercises, as well as supervision in connection to the exercises.

Language of instruction: English

Assessment

The course is examined by an individual written exam carried out in an examination hall at the end of course and assignments normally carried out in groups of 2–3 students. The assignments part is examined on the basis of solutions to compulsory problems handed in during the course and on the basis of individual contribution to the group work.

Students are required to complete written self- and peer-assessment forms during the course which will be part of the assessment of the student's individual contribution to the project.

Retake examinations of the assignments part consist of individual assignments.

If a student, who has failed the same examined component twice, wishes to change examiner before the next examination, a written application shall be sent to the department responsible for the course and shall be granted unless there are special reasons to the contrary (Chapter 6, Section 22 of Higher Education Ordinance).

In cases where a course has been discontinued or has undergone major changes, the student shall normally be guaranteed at least three examination occasions (including the ordinary examination) during a period of at least one year from the last time the course was given.

Grades

The grading scale comprises: Pass with Distinction (VG), Pass (G) and Fail (U).
A Pass grade (G) for the entire course requires at least a Pass grade for all sub-courses.To be awarded Pass with Distinction (VG) for a full course, the student must, in addition, receive a VG on the sub-course written exam.

Course evaluation

The course is evaluated through a meeting after the course between teachers and student representatives. Further, an anonymous questionnaire is used to ensure written information. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed.

Additional information

Course literature to be announced 8 weeks prior to the start of the course.