UNIVERSITY OF GOTHENBURG

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DIT962 Data Structures, 7.5 credits

Datastrukturer, 7,5 högskolepoäng

First Cycle

Confirmation

This course syllabus was confirmed by Department of Computer Science and Engineering on 2021-11-15 and was last revised on 2023-11-09 to be valid from 2025-01-20, spring semester of 2025.

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 Computer Science Bachelor's Programme and 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) Computer Science, Bachelor's Programme (N1COS) and 3) Bachelor's Programme in Mathematics (N1MAT)

Main field of studies	*Specialization*
Computer Science	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 credits imperative or object-oriented programming (e.g., DIT013 Imperative programming with basic object-orientation, or equivalent)
- 7.5 credits in functional programming (e.g., DIT441 Introduction to functional programming, or equivalent)
- 7.5 credits in discrete mathematics (e.g., DITXXX Discrete mathematics for software engineering, DIT984 Discrete mathematics for computer scientists, the sub-course Introductory algebra of MMG200 Mathematics 1, or equivalent)

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, including lists, queues, hash tables, trees, and graphs
- explain some of the algorithms used to manipulate and query these data structures in an efficient way, and explain why they are correct

Competence and skills

- apply basic abstract data types and data structures, and algorithms related to these
- implement and use abstract data types as interfaces, and data structures as classes, in an object-oriented programming language
- read, specify, and describe algorithms, at a higher level of abstraction than code
- implement and use abstract data types and data structures in a functional programming language

Judgement and approach

- analyze the efficiency of basic algorithms and data structures
- make informed choices between different data structures and algorithms for different applications

Course content

Data structures and algorithms are fundamental building blocks in almost all software products. Knowledge and skills in data abstraction, data structures, and algorithms are important in the construction, use, and maintenance of adaptable, reusable, correct, and efficient program components.

The course gives knowledge and skills in the construction and use of algorithms and data structures, an introduction to various techniques for the analysis of algorithms, and insights in the advantages of using data abstraction in program development.

The following topics are covered by the course:

- abstract data types
- data structures and algorithms focusing both on imperative, object-oriented, and functional languages
- differences between lazy and strict evaluation of functional data structures
- common data structures such as arrays, linked lists, unbalanced and balanced trees, heaps, and hash tables
- how these can be used to implement abstract data types such as stacks, queues,

priority queues, maps, sets, and graphs
- standard algorithms for these data structures, including their resource demands
- searching and sorting algorithms
- using different libraries for data structures and algorithms
- basic complexity analysis of data structures and algorithms

Sub-courses

1. **Written hall exam** *(Salsentamen),* 4.5 credits
   Grading scale: Pass with distinction (5), Pass with credit (4), Pass (3) and Fail (U)
2. **Laboratory work** *(Laboration),* 3 credits
   Grading scale: Pass (G) and Fail (U)

Form of teaching

The teaching consists of lectures, exercises and supervised group work.

Language of instruction: Swedish and English
The course is given in Swedish, but English may appear.

Assessment

The course is examined by an individual written exam (4.5 credits), and assignments carried out in groups (3.0 credits).

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 (5), Pass with credit (4), Pass (3) and Fail (U).
The final grade, according to the scale Pass with distinction (5), Pass with credit (4), Pass (3) and Fail (U), is given based on the grade for the written exam.

Course evaluation

The course is evaluated through meetings both during and 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

The course can be taught together with Chalmers.

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

The course replaces the course DIT961 Data structures 7,5 credits. The course cannot be included in a degree which contains DIT961. Neither can the course be included in a degree which is based on another degree in which the course DIT961 is included.