

Data Structures, Algorithms and Programming Paradigms

Programme course

11 credits

Datastrukturer, algoritmer och
programmeringsparadigm

TDDD86

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

Computer Science and Engineering, Computer Science

Course level

First cycle

Advancement level

G1X

Course offered for

- Computer Science and Software Engineering, M Sc in Engineering
- Computer Science and Engineering, M Sc in Engineering

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

Discrete mathematics, and basic knowledge of functional, imperative, and object-oriented programming.

Intended learning outcomes

The purpose of the course is to give the student tools to be able to independently construct computer programs that use time and memory efficiently. Furthermore, the course gives deeper knowledge about programming, specifically procedural and object-oriented programming in the programming language C++, as well as an introduction to programming paradigms as a broader perspective on programming.

Upon completion of this course the student should be able to:

- Demonstrate the ability to analyze time and space complexity of iterative and simple recursive algorithms.

- Explain and use the most common abstract data types and sorting algorithms.
- Implement the most common abstract data types and sorting algorithms with different data structures and algorithms.
- Describe established methods for design (and analysis) of algorithms in general.
- Implement procedural and object-oriented programs in the programming language C++.
- Solve non-trivial computational tasks by using different components of the C++ standard library in combination.
- Describe and compare the most common programming paradigms and their underlying principles.
- Be able to apply effective algorithms to better understand social problems linked to sustainable development.

Course content

- Basic notions
- Mathematical foundations for analysis of algorithms
- Fundamental abstract data types and data structures, such as lists, stacks, queues, search trees, heaps, hash tables, and graphs
- Efficiency analysis of algorithms
- Sorting and searching
- String algorithms
- Graph algorithms
- Algorithmic paradigms (dynamic programming, greedy algorithms, divide and conquer, brute force search)
- Procedural programming in C++ (variables, constants, declarations, expressions, statements, functions, fundamental data types and fundamental data structures)
- Pointers and dynamic memory management
- Object-oriented programming and classes in C++ (class declaration, data members, member functions, access specification for class members, constructors, destructor, inheritance, polymorphism, friends)
- Usage of the C++ standard library (input and output, character and string handling, containers, iterators, algorithms, user defined function objects and lambda expressions)
- Type parameterized functions and classes (templates) in C++
- Introduction to programming paradigms
- Introduction to efficient algorithms importance for society's goals of sustainable development

Teaching and working methods

Lectures are used to present theory that is then tested and implemented in laboratory exercises. The lectures on data structures and algorithms introduce new concepts at a summary level, where after students are expected to obtain detailed knowledge through reading and practicing using a course specific version of the open, interactive, teaching aid OpenDSA.

Examination

DAT1	Computer examination	U, 3, 4, 5	3 credits
LAB1	Programming assignments	U, 3, 4, 5	5 credits
UPG1	Computer based exercises	U, G	2 credits
UPG2	Written assignment	U, G	1 credits

The computer based exercises are performed in a course specific version of the open, interactive, teaching aid OpenDSA.

The computer exam tests the students knowledge about data structures and algorithms. The first part of the examination comprises assignments similar to the computer based exercises, giving a passing grade. For higher grades solving tasks from the second part of the examination is required.

The programming assignments train and test the student's skills and knowledge in programming in C++ and data structures and algorithms. By completing tasks besides those that are mandatory a higher than passing grade can be assigned.

The written assignment trains and tests the student's skills and knowledge in programming paradigms.

The final grade assigned is the weighted average grade of the computer examination and the programming assignments.

Grades

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

Department

Institutionen för datavetenskap

Director of Studies or equivalent

Ahmed Rezine

Examiner

Ahmed Rezine

Course website and other links

<http://www.ida.liu.se/~TDDD86/>

Education components

Preliminary scheduled hours: 90 h

Recommended self-study hours: 203 h

Course literature

Additional literature

Books

Lippman, Lajolie, Moo, (2012) "*C++ Primer*" fifth edition

Websites

Course specific version of OpenDSA.

Compendiums

Lab compendium, C++ style guides, and other material available through the course web pages.

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.