DIT725: Logic, algorithms and data structures, academic year 2013/2014

The course is given in the spring semester at Campus Lindholmen. It's one of the supporting courses in T2 of the SE&M programme. In the course information folder you can find the course syllabus. The course essentially runs in the first half of the spring semester. The course starts on Tuesday, 21 January 2014, at 15:15 in room Alfa. See below for a complete schedule.

Course responsible

Andreas Abel, andreas.abel@gu.se

Supervisors

Khashayar Abdoli, kh.napster@gmail.com
Husam Abdulwahhab, husam.al.azzawi@gmail.com
Einar Sundgren, mail@einarsundgren.se
Gregor Ulm, gregor.ulm@gmail.com

Course Description

The course has three general themes: (1) discrete mathematics and its use for specifying, implementing, and documenting algorithms; (2) the role of algorithms in the design and development of programs; (3) the role of data structures in the implementation of algorithms. These themes are approached laterally such that the relationships between mathematical concepts, the realization of these concepts in a programming language, and the use of these concepts to solve concrete programming problems are highlighted.

These general themes are supported by the study of sub themes from within the fields of mathematics and computer science, all selected to highlight the usefulness of mathematics in programming: formal logic, complexity, sequences, functions and mappings, recursion, set theory, trees, data structures for sequences, data structures for trees, hashtables, orderings, sorting algorithms, graphs, graph algorithms, data structures for graphs, matrices, finite state machines, regular expressions, optimization, data structures for 3D graphics.

Literature

1. Michael T. Goodrich & Roberto Tamassia :
   Data Structures and Algorithms in Java, 5th Ed
   John Wiley & Sons (ISBN: 978-0-470-39880-7)
2. Judith L. Gersting :
   Mathematical Structures for Computer Science, 6th Ed
   W H Freeman & Co (ISBN: 0-7167-6864-X)

Organization

The course consists of 8 mini modules, M1-M8. Each module contains an introductory lecture, supervised exercise session(s), and an assignment (mini problem) which should be handed in the following week. The assignments are done in groups of usually 3 and submitted via GUL. You need to register your group via Communication/Project Groups. Each group will be assigned to one of the supervisors.

The group assignments will be graded with pass or fail. If you fail any of the assignments, you have one more chance to pass them. There is a final re-submission deadline for the assigments towards the end of the semester.

The course finishes with a written exam, and re-exam.

Examination

In order to pass the course you must pass all group assignments and the written exam. You can pass the written exam with either of the grades G and VG. The grade you receive for the whole course is based on the grade of the written exam (provided you have passed all assignments).

Remember that you have to sign-up for the written hall-exam. Under the link below you'll find the Department’s Examschedule and all the needed information for the written hall-exam e.g. dates for the signing-up period and how sign-up and when the exam will take place.

Exam schedule and registration info. (Excerpt: The LAD exam is on 31 March 2014.)

The first re-exam is scheduled for 5 May 2014. Please register!

The second re-exam will take place on **28 August 2014**. Please register!

To practice, look at the exams of 2013. There is also a set of older exams.

The exam of 2014-03-31 and its solution. You can review the grading of your exam in the student office.

The exam of 2014-05-05 and its solution. You can review the grading of your exam in the student office from Thu, 8 May 2014.

Schedule

Lectures take place in lecture room Alfa and are on Tuesdays , starting 15:15, and some Thursdays, starting at 13:00. There are some exceptions, please refer to the schedule below. Group assignments should be handed in on Tuesdays before 8:00 am, see schedule.

See page Mini modules contents for recommended reading and assignments. Note that the deadline of the first assignment has been postponed one week in order for you to have time to form groups. There will also be a number of Q&A sessions after the series of lectures has ended. Come and ask questions about the exam, assignments or anything other related to the course.

Date	Time	Room	What
Tue, Jan 21	15-17	Alfa	M1 lecture
Thu, Jan 23	13-15	Alfa	M1 lecture (ctd.)
Tue, Feb 4	8:00		M1 assignment deadline
Tue, Feb 4	15-17	Alfa	M2 lecture
Thu, Feb 6	13-15	Alfa	M2 lecture (ctd.)
Tue, Feb 11	15-17	Alfa	M3 lecture
Thu, Feb 13	8:00		M2 assignment deadline
Tue, Feb 18	**13-15**	Alfa	M4 lecture
Thu, Feb 20	8:00		M3 assignment deadline
Thu, Feb 20	13-15	Alfa	M5 lecture
Sat, Mar 01	8:00		M4 assignment deadline
Tue, Mar 04	15-17	**Beta**	M6 lecture
Thu, Mar 06	13-15	Alfa	M7 lecture
Sat, Mar 08	8:00		M5 assignment deadline
Sat, Mar 15	8:00		M6 assignment deadline
Thu, Mar 20	13-15	Alfa	M8 lecture
Sat, Mar 22	8:00		M7 assignment deadline
Tue, Mar 25	15-17	Alfa	Q&A session

Supervision

Supervision is provided once a week. Anyone taking the course can consult the assistants at this time. Supervision starts Thu 30 Jan and ends Mon 31 Mar. There are regular supervision times:

Gregor	Thu 15-17	Patricia C
Husam	Fr 10-12	Patricia C?
Einar	Thu 15-17	Patricia C
Khashayar	Fr 10-12	Patricia C?

No supervision on Fr 28 Feb.

If your group needs additional supervision time, please negotiate time and place with your supervisor.

Learning outcomes

- identify and choose algorithms in the engineering research literature, given the nature of a computational problem
- implement such algorithms in a programming language with clean and efficient results
- do simple calculations within the following mathematical theories: statement logic, predicate logic, set theory, graph theory, matrix theory
- read, specify, and describe algorithms, at a higher level of abstractions than code,
- read and follow the mathematical descriptions of algorithms recurrent in engineering research literature,
- present own algorithms using mathematical language and pseudo code,
- improve the performance of inefficient programs, by choice of more appropriate data structures, and algorithms of better complexity,
- reason around and implement data structures such as hash tables, linked lists, binary trees and heaps