Course PM

DIT181, Data Structures and Algorithms, 7.5 Hec, Vt, 2018

Examiner: Gül Calikli (calikli@chalmers.se; gul.calikli@gu.se)
Course Responsibles: Gül Calikli and Michal Palka (michal.palka@chalmers.se)
Teaching Assistants: Snezhina Racheva (gusracsn@student.gu.se), Margit Saal (gussaalma@student.gu.se), Yushu Yu (yushu.yu@chalmers.se), Yue Kang

---

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.
- Algorithm design techniques such as brute force, divide and conquer, transform and conquer, and randomisation.
- Algorithms recurrent in engineering research literature such as searching and sorting.
- Common data structures and abstract data types, such as arrays, lists, maps, sets, trees, hash tables, matrices and graphs.
- Recursion and non-determinism.

Java will be used as an object-oriented programming language for implementing the data structures and algorithms that will be covered in this course.

Sub-courses

1. Written exam, 4.5 higher education credits
   Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)
2. Assignments, 3 higher education credits
   Grading scale: Pass (G) and Fail (U)
   The Assignments will consist of the following:
   - In-Class exercises will make up 10% of the grade for "Assignments".
   - 3 assignments consisting of written (theory and practice related) questions and short programming exercises. (Each assignment makes up 15% of the grade for the sub-course "Assignments", hence making up 45% of the grade for "Assignments" in total.)
   - A programming project together with a written report. (Programming project makes up 45% of the grade for the sub-course "Assignments", 35% being for the source code of the project and 10% being for documentation.)

Learning outcomes:

After completion of the course the student is expected to be able to:

Knowledge and understanding

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

Skills and abilities

- 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

- analyse 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 structure/course implementation:

Getting Prepared for Lectures:
Before coming to the lecture, the students are supposed to read the assigned chapters form their textbook. They are also provided with supplementary tutorial videos, which provide insight about how the algorithm or data structure works by using examples from real life mostly without using code. The students will sometimes be provided with small example codes implemented in Java. They are expected to execute these codes while/after reading the assigned sections from the textbook and watching the assigned videos. In this way, before diving into mathematical and coding related details, the students are introduced with the main idea behind the algorithms and data structures and then they are introduced with further details. This is followed by getting hands dirty by executing provided sample codes. The aim is to help the student learn in an incremental manner without having been exposed to too much cognitive load (i.e., concepts followed by mathematical details followed by implementation-related details).

Lectures:
When the students come to the lecture, there will be a 10-15-minute session during which they will individually answer basic questions that have been prepared on Kahoot, which is a game based learning web platform. These questions aim to measure how much the students got prepared for the course. During the lectures, course responsibles/teachers will conduct ***active learning***: 30-45-minute lecturing will be followed by hands on exercises (pen and paper) done in groups. The answers to these exercises will also be discussed following the exercises if lecture time allows or otherwise during supervision sessions.

Last week's lectures and supervision sessions will be in the form of a flipped classroom. Students will be provided beforehand materials to introduce them with coding interviews at companies such as Facebook, Google, Amazon, Microsoft, etc. Each group will prepare a coding interview question and groups will be paired as one group doing the interview the other being interviewed. They will interview each other on white board and at the end of the interview the others including the lecturer will give feedback by making comments about the interview result.

Examination forms:

The written hall exam will take place on **March 16, 2018**. First and second re-exams will take place on June 5, 2018 and August 28, 2018, respectively.

Course literature:

The following textbook will be used during the lectures:

- Mark A. Weiss, "Data Structures and Problem Solving", Pearson International Edition, Fourth Edition, ISBN-10: 129202576X; ISBN-13: 978-1292025766

The following are recommended reading:

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", The MIT Press, Third Edition (ISBN: 9780262033848)
- Gayle Laakmann McDowell, "Cracking the Code Interview", Career Cup LLC, 6th Edition

Schedule:

The following is the schedule of the lecture and examination sessions.

Date	Time	Theme	Reading Material
16.01.2018	10:15 - 12:00	Introduction, dynamic arrays, binary search	Course Textbook chapter 2, pp. 38-42; chapter 5, pp. 204-206
18.01.2018	10:15 – 12:00	Algorithm Analysis (Big-O notation)	Course Textbook chapter 5, pp. 183-210
23.01.2018	10:15 - 12:00	Sorting algorithms	Course textbook chapter 8, pp. 347-362
24.01.2018	10:15 - 12:00	Supervision Session	
25.01.2018	10:15 - 12:00	Recursion	Course textbook chapter 7, pp. 289-313
26.01.2018	10:15 - 12:00	Supervision Session	
30.01.2018	10:15 - 12:00	Quicksort Algorithm	Course textbook chapter 8, pp. 360-376
31.01.2018	10:15 - 12:00	Supervision Session	
01.02.2018	10:15 - 12:00	Stacks and queues	Course textbook chapter 15, pp. 567-577
02.02.2018	10:15 - 12:00	Supervision Session	
06.02.2018	10:15 - 12:00	Linked Lists, Linked List implementations of stacks and queues	Course textbook chapter 16, pp. 591-618 and chapter 15, pp. 577-586
07.02.2018	10:15 - 12:00	Supervision Session	
08.02.2018	10:15 - 12:00	Trees	Course textbook chapter 17, pp. 623-639
09.02.2018	10:15 - 12:00	Supervision Session	
13.02.2017	10:15 - 12:00	Tree traversal, binary search trees	Course textbook chapter 17, pp. 639-652; chapter 18, pp. 659-678
14.02.2017	10:15 - 12:00	Supervision Session	
15.02.2017	10:15 - 12:00	AVL trees	Course textbook chapter 18, pp. 678-700
16.02.2017	10:15 - 12:00	Supervision Session	
20.02.2017	10:15 - 12:00	Hash Tables	Course textbook chapter 19, pp. 745-772
21.02.2017	10:15 - 12:00	Supervision Session	
22.02.2017	10:15 - 12:00	Priority Queue: The binary heap	Course textbook chapter 20, pp. 779-798
23.02.2017	10:15 - 12:00	Supervision Session	
27.02.2017	10:15 - 12:00	Graphs	Course textbook chapter 13, pp. 501-520

28.02.2017	10:15 - 12:00	Supervision Session	
01.03.2017	10:15 - 12:00	Graphs (continued), Dijkstra's Algorithm	Course textbook chapter 13, pp. 520-526
02.03.2017	10:15 - 12:00	Supervision Session	
06.03.2017	10:15 - 12:00	Course Summary and Coding Interviews	
07.03.2017	10:15 - 12:00	Coding Interviews	
08.03.2017	10:15 - 12:00	Coding Interviews	
09.03.2017	10:15 - 12:00	Coding Interviews	