Course PM

DIT181, Data Structures and Algorithms, 7.5 Hec, Vt, 2019

Examiner and Course Responsible: Gül Calikli (calikli@chalmers.se; gul.calikli@gu.se)

Teachers: Gül Calikli (calikli@chalmers.se; gul.calikli@gu.se), Linda Erlenhov (linda.erlenhov@chalmers.se) and Björnborg Nguyen (bjornborg.nguyen@chalmers.se)

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, and divide and conquer.
- Algorithms recurrent in engineering research literature such as searching and sorting.
- Common data structures and abstract data types, such as arrays, stacks and queues, linked lists, trees and hash tables.
- Recursion.

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)
   - In order to pass with distinction (VG), it is required that the student gets minimum 80 points from the written exam.
   - In order to pass (G), it is required that the student gets minimum 50 points from the written exam.
2. Assignments, 3 higher education credits
   Grading scale: Pass (G) and Fail (U)
   - There will be 3 Assignments, in total.
   - In order to get a Pass (G), it is required that the student gets minimum 50 points from each assignment.
   - Assignments will consist of written (theory and practice related) questions and short programming exercises.
   - Students will work on assignments in groups of 2 (people).
   - There will be skeleton codes provided for each topic in the assignments. We strongly recommend that you use those skeleton codes for completing the programming tasks in assignments, since the test suites we will use for grading is compatible with the provided skeleton codes. We will provide the students the test suites we will use for grading the programming tasks in the assignments, in advance.

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:

The course consists of four components: lectures, hands-on programming sessions, problem sessions and supervision sessions. These components have been designed to asisst you in your learning process (so that learning outcomes are achieved) and eventually succeed in the assignments and written hall exam. The overall design of the course aims 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:
There will be 16 lectures in total. The first lecture will give students details about course design, content and grading, which will be followed by a high-level introduction to data structures and algorithms. The last three lectures will consist of summary of the topics covered in the course and solving some questions as a preparation for the written hall exam (see schedule below). Details about the remaining lectures are given below.

Getting Prepared for Lectures:
Starting from the second lecture, before coming to the lecture, the students are expected to watch tutorial videos and then read the assigned chapters form their textbook. Tutorial videos usually provide insight about how the algorithm or data structure works by using examples from real life mostly without using code. In this way, before diving into mathematical and algorithmic details, the students are introduced with the main idea behind the algorithms and data structures and then they are introduced with further details by reading assigned pages from text book.

During 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. The answers to these questions will be explained by course responsible during the lecture. During the lectures, course responsible will conduct ***active learning***: Teachers' lecturing will be supplemented by hands on in-class exercises (pen and paper) done in groups. Links to Kahoot questions will be published on Canvas after the lecture. Answers to in-class exercises will be published on Canvas on the day after the lecture (see Schedule below for exact times and dates).

Hands-on Programming Sessions:
Hands-on Programming sessions aim to help the students to get prepared for the programming questions in the assignments. The same skeleton code will be used both in the hands-on programming sessions and in the assignments for each topic (e.g., dynamic arrays, sorting, recursion, etc.). At the beginning of each session the teacher who is responsible for the hands-on programming sessions, will explain the skeleton code. This will be followed by a hands-on programming demo conducted by the teacher for the first question. The rest of the session will be conducted employing ***active learning*** approach. The teacher will give a high-level explanation of how students are supposed to do hands-on programming for remaining exercises. Teaching assistants and the teacher will help/guide the students while they are working on their hands-on programming tasks. In case, students might not complete all the tasks, they can continue solving them by themselves after the session. Answers to hands-on programming exercises will be published on Canvas not immediately after the session, but some-time after, in order to give students some time to work on the programming tasks. We designed the hands-on programming tasks in a way such that once the student solves the programming tasks, it will be much easier for them to complete the corresponding assignment questions. Some programming tasks that students will be working on during hands-on programming sessions will be partial or complete answers to some questions in the assignments.

Problem Sessions:
Problem sessions aim to help students to get prepared for solving the theory and practice related non-programming questions in the assignments. The teacher who is responsible for the problem sessions, will solve some questions on white board (in an interactive manner whenever time allows). We prepared the problem session questions in a way such that once the student understands how the questions are solved, it will be much easier for them to complete the corresponding questions in the assignment. Some questions covered during problem sessions will be partial answers to some theoretical questions in the assignments.

Supervision Sessions:
There will be supervision sessions during which students can ask the responsible teacher questions about the assignments they are working on.

Examination forms:

The written hall exam will take place on **March 22, 2019**. First and second re-exams for written hall exam will take place on **June 11, 2019** and **August 27, 2019,** respectively.

Assignments can only be submitted until the deadlines. Late submissions will not be accepted. See the course schedule below for the deadlines (e.g., dates and times) of each assignment. As a part of the re-examination period, deadline for re-submission of the assignments is **April 25,**

2019. Students can only re-submit the assignments, which they failed or could not submit until the assignment deadlines that are indicated in the course schedule below.

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

Course Schedule:

See Canvas for a more detailed version of the schedule, which includes information about assigned reading material and tutorial videos for each lecture. (see next page)

Date	Time	Theme	Responsible	Location
22.01.2019	10:15 - 12:00	Lecture #1: Kick-off and Introduction	Gül Calikli	Alfons
23.01.2019	08:15 - 10:00	No Hands-on Programming Session		
24.01.2019	10:15 – 12:00	Lecture #2: Dynamic arrays	Gül Calikli	Alfons
24.01.2019	10:00	Deadline for formation of groups for Assignments	Students	
25.01.2019	08:15 - 10:00	Hands on Programming Session #1 (dynamic arrays) Groups: #1-#25	Linda Erlenhov	Mållgan
25.01.2019	10:15 - 12:00	Hands on Programming Session #1 (dynamic arrays) Groups: #26-#50	Linda Erlenhov	Mållgan
25.01.2019		Recommendation to students: Start Assignment #1- You should be able to solve questions 1-4)		
25.01.2019	08:00	Solutions to in-class exercises for Lecture #2 (dynamic arrays) will be published on Canvas platform	Gül Calikli	
25.01.2019	18:00	Solutions to Hands-on Programming Session #1 on dynamic arrays will be published on Canvas platform	Linda Erlenhov	
END OF WEEK #1				
29.01.2019	10:15 – 12:00	Lecture #3: Algorithm Analysis (Big-O notation)	Gül Calikli	Alfons

30.01.2019 10:15 - 12:00	Supervision Session #1 (for instance, students can ask questions about Assignment #1 questions 1-4)	Björnborg Nguyen	Mållgan
31.01.2019 10:15 - 12:00	Lecture #4: Algorithm Analysis (Big-O notation)	Gül Calikli	Alfons
31.01.2019 10:00	Test suite for ASSIGNMENT#1 will be published on Canvas.	Björnborg Nguyen	
01.02.2019 08:15 - 10:00	Problem Session #1 Algorithm Analysis (Big-O notation) Groups: #26-#50	Gül Calikli	Mållgan
01.02.2019 10:15 - 12:00	Problem Session #1 Algorithm Analysis (Big-O notation) Groups: #1-#25	Gül Calikli	Mållgan
01.02.2019 08:00	Solutions to In-Class Exercises in Lectures #3 and #4 (algorithm analysis, Big-O notation) will be published on Canvas platform	Gül Calikli	
01.02.2019	Recommendation to students: Start Big-O analysis questions in Assignment #1- You should be able to solve Big-O analysis questions)		
END OF WEEK #2			
05.02.2019 10:15 - 12:00	Lecture #5: Sorting algorithms	Gül Calikli	Alfons
06.02.2019 10:00	Solutions to Problem Session #1 (on algorithm analysis, Big-O notation) will be published on Canvas platform	Gül Calikli	
06.02.2019 08:15 - 10:00	Hands on Programming Session #2 (sorting) Groups: #1-#25	Linda Erlenhov	Mållgan

06.02.2019 10:15 - 12:00	Hands on Programming Session #2 (sorting) Groups: #26-#50	Linda Erlenhov	Mållgan
06.02.2019	Recommendation to students: Start sorting question in Assignment #1- You should be able to solve sorting question except for implementation of quicksort algorithm)		
06.02.2019 15:15-17:00	Supervision Session #2 (for Assignment #1)	Björnborg Nguyen	Mållgan
07.02.2019 10:15 - 12:00	Lecture #6: Recursion	Gül Calikli	Alfons
08.02.2019 08:00	Solutions to In-Class Exercises for Lecture #6 on recursion will be published on Canvas	Gül Calikli	
08.02.2019 08:15 - 10:00	Problem Session #2 (Recursion) Groups: #26-#50	Gül Calikli	Mållgan
08.02.2019 10:15 - 12:00	Problem Session #2 (Recursion) Groups: #1-#25	Gül Calikli	Mållgan
08.02.2019 15:15 - 17:00	Supervision Session #3 (for Assignment #1)	Björnborg Nguyen	Mållgan
END OF WEEK #3			
11.02.2019 08:00	**Deadline for ASSIGNMENT #1 (dynamic arrays, complexity analysis – including Big-O notation)**		
11.02.2019 10:00	Solutions to ASSIGNMENT#1 will be published on Canvas	Björnborg Nguyen	
11.02.2019 10:00	Solutions to Problem Session #2 (on recursion) will be published on Canvas	Gül Calikli	

12.02.2019 10:15 - 12:00	Lecture #7: Recursion	Gül Calikli	Alfons
13.02.2019 08:00	Solutions to In-Class Exercises for Lecture #7 (on recursion) will be published on Canvas	Gül Calikli	
13.02.2019 08:15 - 10:00	Hands on Programming Session #3 (Recursion) Groups: #1-#25	Linda Erlenhov	Mållgan
13.02.2019 10:15 - 12:00	Hands on Programming Session #3 (Recursion) Groups: #26-#50	Linda Erlenhov	Mållgan
13.02.2019	Recommendation to students: Start Assignment #2- You should be able to solve recursion questions		
14.02.2019 10:15 - 12:00	Lecture #8: Quicksort Algorithm	Gül Calikli	Alfons
14.02.2019 10:00	Test suite for ASSIGNMENT#2 will be published on Canvas	Björnborg Nguyen	
15.02.2019 08:00	Solutions to In-Class Exercises for Lecture #8 (quicksort algorithm) will be published on Canvas	Gül Calikli	
15.02.2019 08:15 - 10:00	Hands on Programming Session #4 (Quicksort Algorithm) Groups: #26-#50	Linda Erlenhov	Mållgan
15.02.2019 10:15 - 12:00	Hands on Programming Session #4 (Quicksort Algorithm)	Linda Erlenhov	Mållgan

	Groups: #1-#25		
15.02.2019 14:00	Solutions to Hands-on Programming Session #3 (recursion) will be published on Canvas	Linda Erlenhov	
15.02.2019	Recommendation to students: Start Assignment #2- You should be able to solve quicksort algorithm)		
15.02.2019 15:15 - 17:00	Supervision Session #4 (for Assignment#2)	Björnborg Nyugen	
15.02.2019 10:00	Grades for ASSIGNMENT#1 will be published on Canvas		
END OF WEEK #4			
19.02.2019 10:15 - 12:00	Lecture #9: -Stacks and queues	Gül Calikli	Alfons
20.02.2019 08:00	Solutions to In-Class Exercises for Lecture #9 (on stacks and queues) will be published on Canvas	Gül Calikli	
20.02.2019 08:15 - 10:00	Hands on Programming Session #5 (Stacks and queues) Groups: #1-#25	Linda Erlenhov	Mållgan
20.02.2019 10:15 - 12:00	Hands on Programming Session #5 (Stacks and queues) Groups: #26-#50	Linda Erlenhov	Mållgan
20.02.2019 15:15 - 17:00	Supervision Session #5 (for Assignment#2)	Björnborg Nguyen	
21.02.2019 10:15 - 12:00	Lecture #10: Linked Lists	Gül Calikli	Alfons

22.02.2019 08:00	Solutions for In-Class Exercises for Lecture #10 (for linked lists) will be published on Canvas	Gül Calikli	
22.02.2019 08:15 - 10:00	Hands on Programming Session #6 (Linked Lists) Groups: #26-#50	Linda Erlenhov	Mållgan
22.02.2019 10:15 - 12:00	Hands on Programming Session #6 (Linked Lists) Groups: #1-#25	Linda Erlenhov	Mållgan
22.02.2019 15:15 - 17:00	Supervision Session #6 (for Assignment#2)	Björnborg Nguyen	
22.02.2019 10:00	Solutions of Hands-on Programming Session #5 (for stacks and queues) will be published on Canvas	Linda Erlenhov	
END OF WEEK #5			
25.02.2019 08:00	**Deadline for ASSIGNMENT #2 (recursion, sorting algorithms including quicksort algorithm)**		
25.02.2019 10:00	Solutions to ASSIGNMENT#2 will be published on Canvas	Björnborg Nguyen	
26.02.2019 10:15 - 12:00	Lecture #11: Trees	Gül Calikli	Alfons
27.02.2019	Recommendation to students: Start Assignment #2- You should be able to solve quicksort algorithm)		
27.02.2019 10:00	Solutions of Hands-on Programming Session #6 (for linked lists) will be published on Canvas	Linda Erlenhov	

27.02.2019 10:15 - 12:00	No supervision or hands-on programming session		
28.02.2019 10:15 - 12:00	Lecture #12: Trees	Gül Calikli	Alfons
28.02.2019 10:00	Test suites for ASSIGNMENT#3 will be published on Canvas	Björnborg Nguyen	
01.03.2019 08:00	Solutions for In-Class Exercises for Lectures #11 and #12 (for trees) will be published on Canvas	Gül Calikli	
01.03.2019 08:15 - 10:00	Hands on Programming Session #7 (Trees) Groups: #1-#25	Linda Erlenhov	Mållgan
01.03.2019 10:15 - 12:00	Hands on Programming Session #7 (Trees) Groups: #26-#50	Linda Erlenhov	Mållgan
01.03.2019	Recommendation to students: Start Assignment #3- You should be able to solve stacks and queues questions)		
01.03.2019 17:00	Grades for ASSIGNMENT#2 will be published on Canvas	Björnborg Nguyen	
END OF WEEK #6			
05.03.2019 08:00	Solutions of Hands on Programming Session #7 (for trees) will be published on Canvas	Linda Erlenhov	
05.03.2019 10:15 - 12:00	Lecture #13: Hash Tables	Gül Calikli	Alfons
06.03.2019 08:00	Solutions for In-Class Exercises for Lecture #13 (hash tables) will be published on Canvas	Gül Calikli	

06.03.2019 08:15 - 10:00	Hands on Programming Session #8 (Hash Tables) Groups: #26-#50	Linda Erlenhov	Mållgan
06.03.2019 10:15 - 12:00	Hands on Programming Session #8 (Hash Tables) Groups: #1-#25	Linda Erlenhov	Mållgan
06.03.2019 15:15 - 17:00	Supervision Session #7 (for Assignment #3)	Björnborg Nguyen	Mållgan
06.03.2019	Recommendation to students: Start Assignment #3- You should be able to solve hash table question)		
07.03.2019 10:15 - 12:00	Lecture#14: Review for Written Hall Exam	Gül Calikli	Alfons
08.03.2019 10:15 - 12:00	Supervision Session #8 (for Assignment #3)	Björnborg Nguyen	Mållgan
END OF WEEK #7			
11.03.2019 08:00	**Deadline for ASSIGNMENT #3 (stacks and queues, linked lists, trees) (also includes BONUS question about hash tables)**		
11.03.2019 10:00	Solutions to ASSIGNMENT#3 will be published on Canvas	Björnborg Nguyen	
12.03.2019 10:15 - 12:00	Lecture #15: Review for Written Hall Exam	Gül Calikli	Alfons
13.03.2019 10:15 - 12:00	Supervision Session #9 (Q&A session for Written Hall Exam)	Björnborg Nguyen	Mållgan
14.03.2019 10:15 - 12:00	Lecture #16: Review for Written Hall Exam	Gül Calikli	Alfons
15.03.2019 10:15 - 12:00	No supervision or hands-on programming session		

15.03.2019 18:00	Grades for ASSIGNMENT#3 will be published	Björnborg Nguyen	
22.03.2019	**WRITTEN HALL EXAM**		
END OF WEEK #8			