Detailed Course Description - Course Plan Development and Updating Procedures/ Department of Software Engineering	QF01/0408-3.0E

Faculty	Science & I.T.	Department	**Software Engineering**
Course number	0114313	Course title	Algorithms
Number of credit hours	3	Pre-requisite/co-requisite	0114212

Brief course description

This course aims to introduce the concepts of algorithm design and analysis. Its topics cover the following concepts: Solving summations and recurrences. Efficiency and complexity analysis. Tree terminology and algorithms. Binary trees. Hashing methods and solving collision in hashing. Heaps and heap sort. Insertion sort, merge sort and quicksort. Graph terminology, representation and algorithms. Algorithms of Prim, Kruskal, Dijkstra and Floyd. Breadth-first and depth-first search. The greedy, divide-and-conquer, and dynamic programming techniques.

	Course goals and learning outcomes
Goal 1	Ability to use the principles of computer science in understanding, implantation and analysis of mathematical problems and finding their solutions.
Learning outcomes	1.1 Student should understand and analyze mathematical problems. 1.2 Student should be able to use mathematical concepts in algorithm analysis. 1.3 Designing and applying basic algorithms for solving graph problems.
Goal 2	Ability to analyze, design and implement efficient and reliable computer programs.
Learning outcomes	2.1 Student should know different programming methods. 2.2 Analyzing efficiency of algorithms and comparing time and memory efficiency of different algorithms. 2.3 Critical analysis of problems, and then choosing appropriate data structures and designing algorithms for solving the problems discussed and other related problems.
Goal 3	Ability to employ different common algorithms in solving different problems.
Learning outcomes	3.1 Applying basic searching and sorting algorithms. 3.2 Applying hashing methods for data storage and retrieval. 3.3 Performing graph optimization using common algorithms.
Goal 4	Ability to employ complexity theory in algorithm analysis.
Learning outcomes	4.1 Student should understand how some problems are not solvable. 4.2 Student should differentiate between complexity classes and identify common problems that belong to each class. 4.3 Student should classify which problems can be solved in feasible time based on their complexity and applications.
Textbook	1. Anany Levitin, *Introduction to the Design and Analysis of Algorithms*, 3rd ed., 2012. (*Main textbook used in the course timeline*) 2. Heman Jain, *Problem Solving in Data Structures & Algorithms Using JAVA*, 1st Ed., 2017.
Supplementary references	1. Jay Wengrow, *A Common-Sense Guide to Data Structures and Algorithms: Level Up Your Core Programming Skills*, 1st Ed., 2017. 2. Tim Roughgarden, *Algorithms Illuminated: Part 1: The Basics*, 1st Ed., 2017. 3. George T. Heineman, Gary Pollice and Stanley Selkow, *Algorithms in a Nutshell: A Practical Guide*, 2nd Ed, 2016.

Course timeline				
Week	**Number of hours**	**Course topics**	**Pages (textbook)**	**Notes**
01	**1** **1** **1**	**Introduction**: specifications of an algorithm, mathematical background. **Time and memory efficiency of algorithms:** analyzing time and memory requirements of algorithms.	3-39, 475-491	
02	**1** **1** **1**	**Efficiency levels of algorithms:** asymptotic growth rates of functions, formal definitions of Big-O, Ω and Θ classes, analysis of iterative algorithms, analysis of recursive algorithms.	41-95	
03	**1** **1** **1**	**Searching unsorted and sorted lists:** brute force method, linear search, its worst- and average-case time efficiency, binary search and its analysis.	61-98, 104-106, 150-152	
04	**1** **1** **1**	**Searching unsorted and sorted lists:** solving recursive equations, optimal algorithms, optimality of linear and binary search algorithms for unsorted and sorted lists.	61-95, 150-152, 475-491	
05	**1** **1** **1**	**Hashing method:** hash tables, hash functions, resolving collisions, searching, reading and writing in hash tables. **Simple sorting algorithms**: selection sort and its analysis.	269-276 98-100	
06	**1** **1** **1**	**Simple sorting algorithms**: insertion sort and its analysis. Review of the previous topics, solutions of problems. **First Exam.**	131-138	
07	**1** **1** **1**	**Divide and conquer technique:** recursive sorting algorithms, merging of sorted lists and its analysis, mergesort algorithm, tracing and analysis of mergesort.	169-175	
08	**1** **1** **1**	**Divide and conquer technique:** quicksort idea, quicksort algorithm, tracing and analysis of quicksort.	176-182	
09	**1** **1** **1**	**Graphs and trees:** graph types and their static and dynamic representations, space complexity.	25-39, 182-186	
10	**1** **1** **1**	**Heapsort:** min- and max-heaps, properties, representation, heapsort algorithm, tracing and analysis of heapsort.	226-234	
11	**1** **1**	**Graph traversals:** depth-first search and breadth-first search of graphs.	122-130	

	1	Review of the previous topics, solution of problems.		
12	**1** **1** **1**	**Second Exam.** **Greedy algorithms for graph problems:** greedy technique, minimum spanning tree (MST), Prim's MST algorithm, Kruskal's MST algorithm, analysis of MST algorithms.	315-333	
13	**1** **1** **1**	**Greedy algorithms for graph problems:** shortest paths in graphs, Dijkstra's shortest-path algorithm.	333-338	
14	**1** **1** **1**	**Dynamic programming technique:** dynamic programming concept, Fibonacci numbers example. Warshall-Floyd algorithm for transitive closure.	283-313	
15	**1** **1** **1**	**Computational complexity:** Classes of complexity: P, NP, NP-hard, NP-complete, examples of NP-complete problems. Discussion of assignments, general review of the course topics, solutions of problems.	401-411	
16	**1** **1**	**Final Exam**		

Theoretical course evaluation methods and weight	Participation = 10% First exam 20% Second exam 20% Final exam 50%	**Practical (clinical) course evaluation methods**	Semester students' work = 50% (Reports, research, quizzes, etc.) Final exam = 50%

Approved by head of department		**Date of approval**	

Extra information (to be updated every semester by corresponding faculty member)

Name of teacher		Office Number	
Phone number (extension)		Email	
Office hours			