I053	**Data Structures and Algorithms I**	L	S	E	ECTS
		3	0	2	7

Course objectives. The main objective of the course is to introduce elementary data structures and algorithmic techniques with their applications to solving different computational problems. An efficient implementation of data structures and algorithms, by using some programming language, is also one of the most important objectives.

Course prerequisites. Introduction to Computer Science. Object-oriented programming.

Syllabus.

1. **Introduction.** Basic terms and definitions. Design and analysis of algorithms. Example: Insertion sort and Merge sort.
2. **Asymptotic notation.** Recurrences. The substitution method for solving recurrences. The recursion-tree method for solving recurrences. The master method for solving recurrences.
3. **Divide-and-Conquer.** Binary search. The algorithm for efficient exponentiation. Strassen's algorithm for matrix multiplication.
4. **Randomized algorithms**.
5. **Sorting algorithms and Order Statistics.** Bubble sort. Heapsort. Quicksort. A randomized version of quick sort. Sorting in linear time. Order statistics.
6. **Elementary Data Structures.** Stacks and queues. Linked lists. Trees. Hash tables and associative arrays.
7. **Binary Search Trees.** Querying a binary search tree. Insertion and deletion.
8. **Priority Queues.**
9. **Greedy Algorithms.** Elements of greedy strategy. Applications in different computational problems.
10. **Dynamic Programming.** Elements of dynamic programming. Applications in different computational problems.
11. **Amortized Analysis.**

EXPECTED LEARNING OUTCOMES

No.	LEARNING OUTCOMES
1.	To understand the basics of data structures and algorithms.
2.	To carry out the analysis of correctness, time and space complexity of the algorithms written in pseudo-code.
3.	To use the appropriate data structure to efficiently solve algorithmic problems.
4.	Understanding and application of the following algorithmic techniques: divide-and-conquer, greedy strategy and dynamical programming.
5.	To efficiently implement algorithms in different programming languages.

COUPLING OF THE EXPECTED LEARNING OUTCOMES, TEACHING PROCESS ORGANIZATION AND THE EVALUATION OF THE TEACHING OUTCOMES

	ECTS				SCORE

TEACHING PROCESS ORGANIZATION		EXPECTED LEARNING OUTCOMES **	STUDENT ACTIVITY *	EVALUATION METHOD	min	max
Lecture attendance	1	1-5	Class atendance, discussion, solving the problems individually and in a team	Lists with signatures, observing the activity during the lectures	0	10
Homework	1	1-4	Solving the problems individually	Grading	17	30
Repeated exams	2	1-4	Preparation for the written exam	Grading	17	30
Final exam	2	1-4	Revising	Oral exam	16	30
TOTAL	7				50	100

Teaching methods and student assessment. Lectures contain a deep and systematic overview of elementary data structures and algorithms. During exercises students are expected to solve given programming problems by using acquired knowledge. The correctness and time and space complexity of implemented algorithms are the most important elements. At the end of each practice session students individually solve short quizzes. During the semester, students solve homework assignments that contain programming problems. The assessment of theoretical knowledge is done by written examinations. If students achieve satisfactory results in homework and written exams, they are not obliged to take final written and oral exams.

Can the course be taught in English: Yes

Basic literature:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algoritms, 3Ed, MIT Press, 2009.

Recommended literature:

1. R. Sedgewick, Algorithms in C++, Parts 1-4 Fundamentals, Data Structure, Sorting, Searching, Third Edition, 1998.
2. B. Motik, J. Šribar: Demistificirani C++, Element, Zagreb, 1997.
3. M. T. Goodrich, R. Tamassia, D. M. Mount, Data Structures and Algorithms in C++, Wiley, 2010.
4. A. Drozdek, Data Structures and Algorithms in C++, Cengage Learning, 2012.
5. R. Sedgewick, K. Wayne, Algorithms, Addison-Wesley Professional, 2011.
6. M. J. Atallah, Algorithms and Theory of Computation Handbook, CRC Press, 1998.