

I063	Computational-Thinking And Programming II	L	P	S	ECTS 5
		2	2	0	

Course objectives. The main aim of this course is to familiarize students with the basics of data structures and algorithms and their application in different situations. All algorithms will be implemented in C++ programming language. In addition, students will be able to write algorithms in object-oriented paradigm.

Prerequisites. Computational-Thinking And Programming I.

Course content.

1. Introduction. Basic concepts and definitions. Types of data and commands from elementary to complex data structures - from commands to functions and programs. Abstract structure. Algorithms. Asymptotic analysis.
2. Solving recursion.
3. Sequential and binary search. Procedures sorting: bubble sort, insertion sort, heap sort, selection-sort, quick sort and others.
4. A divide-and-conquer algorithms: N-th power of. N-th Fibonacci number. Merge sort.
5. Linear data structures: Arrays. Lists (single and double linked, circular). Queues. Stacks.
6. Non-linear data structures: trees. The representation of the tree using the array. Tree search. Heap. Binary tree search (Binary Tree Search). Graphs. Graph traversal.

LEARNING OUTCOMES

No.	LEARNING OUTCOMES
1.	Use divide-and-conquer algorithm in the given problems.
2.	Develop basic algorithms for searching and sorting in the field.
3.	Construct a data structure tree, and create algorithms for traversal, search, adding and deleting the tops of trees.
4.	Analyze the running time and the efficiency of memory usage for basic sorting algorithms in the field and a tree.
5.	Analyze the operation of the algorithm search in width and depth in a given graph.

RELATING THE LEARNING OUTCOMES, ORGANIZATION OF THE EDUCATIONAL PROCESS AND ASSESSMENT OF THE LEARNING OUTCOMES

TEACHING ACTIVITY	ECTS	LEARNING OUTCOME **	STUDENT ACTIVITY*	EVALUATION METHOD	POINTS	
					min	max
Attending lectures and exercises	1	1-5	The presence at lectures, discussions, teamwork and independent work on assignments	Attendance lists, tracking activities	3	10
Assignments	2	1-5	Solving problems	Verification of correct answers (evaluation)	17	30

Written exam (Mid-terms)	2	1-5	Preparing for written exams	Verification of correct answers (evaluation)	30	60
TOTAL	5				50	100

Teaching methods and knowledge assessment. In lectures, students will learn programming with a focus on the implementation of elementary data structures and algorithms. In exercises, students solve programming tasks where they use of the basic data structures. In programming tasks, emphasis is on fairness and efficiency of the implemented algorithm. During the semester, students solve assignments, which consist of programming tasks. The theoretical and practical knowledge will be examined in the written exam. Satisfactory scores on programming assignments and quizzes can replace written exam.

Can the course be taught in English: Yes

Basic literature:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 3Ed, MIT Press, 2009.
2. S. Grabusin, Lj. Miletić, Zbirka riješenih zadataka iz C programskog jezika, Pentium, 1996.

Recommended literature:

1. R. Sedgewick, Algorithms in C++, Parts 1-4 Fundamentals, Data Structure, Sorting, Searching, Third Edition, 1998.
2. J. Šribar, B. Motik: Demistificirani C++, 4. dopunjeno izdanje usklađeno sa standardom C++11/C++14, Element, Zagreb, 2014.
3. M. T. Goodrich, R. Tamassia, D. M. Mount, Data Structures and Algorithms in C++, Wiley, 2010.
4. M. J. Atallah, Algorithms and Theory of Computation Handbook, CRC Press, 1998.