COURSE UNIT (MODULE) DESCRIPTION

Course unit (module) title	Code
Data structures and algorithms	

Academic staff	Core academic unit(s)
Coordinating: dr. V. Giedrimas **Other: dr. M. Stoncelis, prof. Dr. D. Dzemydiene**	Šiauliai Academy

Study cycle	Type of the course unit
Bachelor	Mandatory

Mode of delivery	Semester or period when it is delivered	Language of instruction
Auditorium	2nd semester	Lithuanian/ English

Requisites	
Prerequisites: Discrete structures.	**Co-requisites (if relevant):**

Number of ECTS credits allocated	Student's workload (total)	Contact hours	Individual work
5	133	48	85

Purpose of the course unit

To introduce the theoretical foundations of informatics science, to teach understanding and analysis of complex tasks, to teach independent application of non-trivial theoretical knowledge. To provide basic knowledge about data structures, their types and use cases, to teach how to select optimal data structures. Cultivated competences:

- BK1 Application of knowledge
- BK2 Continuous learning
- BK3 Personal abilities
- DK2 Abilities to conduct program system research
- DK3 PS Special Abilities
- DK4 Knowledge and abilities of conceptual foundations

Learning outcomes of the course unit	Teaching and learning methods	Assessment methods
Will know classic data structures and their essential properties.	Case analysis (case studies), Interactive lecture	Exam, defense of laboratory work.
Will be able to select data structures for the task to be solved, compare them, assess suitability for the solution of a specific task.	Case analysis (case studies), Interactive lecture	Exam, defense of laboratory work.
Able to implement methods of data organization and storage in external media	Case analysis (case studies), Interactive lecture	Exam, defense of laboratory work.
Demonstrate knowledge of the theoretical foundations of computer science: be able to indicate the main ways of formalizing algorithms and their influence on computer science, name how the	Problem teaching, interpretation, Case analysis (case studies), group discussion	Written survey, proving theorems, solving problems by applying the theory, mini-tests

complexity of tasks is assessed, discuss solvable and unsolvable problems.		
Will understand formal information: will be able to read and absorb information that is presented using formal methods and various notations.		
Will be able to analyze and evaluate tasks and their solutions: determine the solvability of the task, distinguish between the complexity of the task and the algorithm that solves it, determine the complexity of the tasks and their possible solutions in terms of memory and time and compare them.		
Will be able to apply the knowledge of the theory of algorithms: use statements in problem solving, draw conclusions in specific situations and argue them with theoretical knowledge.	Problem teaching, Case analysis (case studies), Interactive lecture, group discussion	
Will be able to explain proofs of theorems.	Case analysis (case studies), Interactive lecture	Solving standard problems, solving problems using theory, mini-tests
Will be able to solve various problems of algorithm theory: Turing machines, finite automata, recursive functions.		
Will be able to apply different calculations used in propositional logic.		

Content	**Contact hours**							**Individual work: time and assignments**	
	Lectures	Tutorials	Seminars	Workshops	Laboratory work	Internship	**Contact hours, total**	**Individual work**	**Tasks for individual work**
Algorithm complexity analysis. Asymptotic estimates of algorithm complexity. Standard classes of algorithms: P, NP, CNP.	4				2		6	8	Defense of laboratory work, Exam
Sorting algorithms. Selection, insertion and bubble methods.	1				2		3	5	Defense of laboratory work, Exam
Fast sorting algorithms.	1				2		3	5	Defense of laboratory work, Exam
Algorithm strategies: brute force, divide and rule, greedy, return.	2				2		4	5	Defense of laboratory work, Exam
Graphs. Shortest path algorithms: Dijkstra, Belman and Ford, Floyd.	1				2		3	5	Defense of laboratory work, Exam
Algorithms for finding the minimum spanning tree: Primo, Kruskal and Boruvka.	1				2		3	5	Defense of laboratory work, Exam
Algorithms for finding the maximum flow in networks: Ford and Falkerson, Edmond and Karp. Algorithm complexity analysis.	2				2		4	5	Defense of laboratory work, Exam
Algorithms for generating parts. Algorithm complexity analysis.	1				2		3	5	Defense of laboratory work, Exam

Algorithms for generating combinations and sequences. Algorithm complexity analysis.	1				2		3	5	Defense of laboratory work, Exam
Concept of data structure. Overview of data structures.	1				0		1	5	Exam
Linear dynamic structures.	2				2		4	5	Defense of laboratory work, Exam
Hierarchical data structures.	6				2		8	5	Defense of laboratory work, Exam
Data storage on external media. JSON and XML.	1				2		3	5	Defense of laboratory work, Exam
Exam and preparation								17	
Total	**24**				**24**		**48**	**85**	

Assessment strategy	Weight %	Deadline	Assessment criteria
Defense of laboratory work	30%	Every week	Laboratory works and their defense are evaluated. A total of 12 laboratory works
Homework	20%	Penultimate week	Students write an algorithm that examines the program, perform a theoretical analysis, present their work, and answer questions.
Control work	25	Mid-semester	Written practical tasks from some topics are evaluated separately.
Exam	25%	During the exam session	Theoretical questions from the entire course are evaluated separately.

Author (-s)	Publishing year	Title	Issue of a periodical or volume of a publication	Publishing house or web link
Required reading				
Malhotra, D., Malhotra, N.	2020	Data Structures and Program Design Using JAVA.		Dulles:Mercury Learning And Information
Cormen, T., Leiserson, C., Rivest, R., Stein, C.	2001	Introduction to Algorithms		London: McGraw-Hill Company
Recommended reading				
Brass, P.	2008	Advanced Data Structures		Cambridge:Cambridge University Press