COURSE UNIT (MODULE) DESCRIPTION

Course unit (module) title	Code
Algorithms and Data Structures / Algoritmai ir duomenų struktūros	**DSTADS**

Academic staff	Core academic unit(s)
Coordinating: Assist. Prof. Dr. Margarita Beniušė **Other:** Lect. Austėja Bauraitė	Vilnius University Faculty of Mathematics and Informatics, Naugarduko g. 24, LT-03225, Vilnius, tel. +370 5 219 3050, e-mail studijos@mif.vu.lt

Study cycle	Type of the course unit
1st (bachelor)	Compulsory / optional for individual studies

Mode of delivery	Semester or period when it is delivered	Language of instruction
Face to face	2nd semester	English

Requisites	
Prerequisites: ≥ B2 level of English	**Co-requisites (if relevant):**

Number of ECTS credits allocated	Student's workload (total)	Contact hours	Individual work
5	133	64	69

Purpose of the course unit		
The purpose of this course is to develop students' ability to understand an imperative programming paradigm and design simple algorithms, enabling them to solve computational problems and develop computational thinking skills.		
Learning outcomes of the course unit	**Teaching and learning methods**	**Assessment methods**
After completing the course, students will be able to:		
DK.1.1 and **DK.1.2** • explain the fundamental concepts, logic, and principles of algorithms and data structures, and their relevance to computational problem-solving	Case studies, demonstrations, discussions, practical exercises, consultations, literature analysis	Written exam
DK.3.5 • identify and select appropriate data structures and data types most suited within the defined context (problem) for the designed algorithm to ensure its efficiency and applicability	Case studies, discussions, continuous practical exercises, literature analysis, pseudo-code iteration tasks, visualizations, interactive group work	Practical quizzes, written exam
DK.3.6 • design algorithms, define the input and output requirements, and represent the algorithms using various notations and approaches	Case studies, showcasing, discussions, practical continuous exercises, literature analysis, visualizations	Practical quizzes, written exam
BK.4.4 • identify problems within a given context, evaluate potential	Case studies, discussions, consultations, literature analysis	Practical quizzes, written exam

algorithmic solutions, and justify the choice of algorithms and data structures		

Content	Contact hours							Individual work: time and assignments	
	Lectures	Tutorials	Seminars	Workshops	Laboratory work	Internship	**Contact hours, total**	**Individual work**	**Tasks for individual work**
Introduction to subject and computational thinking.	2				2		**4**	**1**	Assigned readings, practical exercises.
Computer architecture and data allocation in memory.	2				2		**4**	**2**	Assigned readings and visualizations; practical exercises.
Abstract data types and data types in programming languages. Showcasing examples from different programming languages, including Python context.	2				2		**4**	**4**	Assigned readings, analysis of examples, practical exercises, including experimentation with programming languages on online platforms.
Basic constructs to design algorithms. Quiz.	2				4		**6**	**6**	Assigned readings, practical exercises covering understanding of basic constructs.
Algorithm representation using various approaches.	4				2		**6**	**10**	Assigned readings, practical exercises, and case study; building an algorithm and presenting using various approaches.
Defining the input and output and iterating the algorithm. Quiz.	2				4		**6**	**8**	Assigned readings, practical exercises.
Arrays and dynamic structures. Pointers. Showcasing from different programming languages. Quiz.	4				2		**6**	**8**	Assigned readings and visualizations, practical exercises, and case study.
FIFO and LIFO principles. Quiz.	2				2		**4**	**4**	Assigned readings, practical exercises.
Dynamic sets.	2				2		**4**	**4**	Assigned readings, practical exercises.
Hierarchical data structures. Quiz.	2				2		**4**	**6**	Assigned readings and visualizations, and practical exercises. Interactive group work.
Hash tables and hash algorithms. Quiz.	2				2		**4**	**4**	Assigned readings, practical exercises.
Text search algorithms.	4				4		**8**	**8**	Assigned readings, practical exercises. Interactive group work.

Algorithmic analysis.	2				2		**4**	**4**	Assigned readings, practical exercises.
Total	**32**				**32**		**64**	**69**	

Assessment strategy	Weight %	Deadline	Assessment criteria
Practical quizzes: average grade	50%	During the semester	Each practical test consists of 2–4 tasks and exercises (in hybrid approach: partially online, partially on paper); tasks and exercises of the test have different weights. Students should demonstrate computational thinking skills related to algorithm design, data types, and algorithm interpretation. The correctness of answers and solutions are assessed. Test assessment scores make an accumulative mark. Yet, students must reach at least 70% of the score within each test to gain admission to the examination. Each student has a second attempt option at each test after failing the first attempt. Ten-point assessment scale: 10 (excellent). Excellent, exceptional knowledge and abilities. 95-100% of questions answered correctly. 9 (very good). Very good knowledge and abilities. 85-94 % of questions answered correctly. 8 (good). Knowledge and abilities are above average. 75-84 % of questions answered correctly. 7 (average). Average knowledge and abilities; there are a few non-essential mistakes. 65-74 % of questions answered correctly. 6 (satisfactory). Knowledge and abilities are below average; there are mistakes. 55-64 % of questions answered correctly. 5 (weak). Knowledge and abilities meet the minimum requirements. 51-54 % of questions answered correctly. 4, 3, 2, 1. Minimum requirements are not met to varying extents.
Written exam	50%	Examination period	In written examination, a set of questions of various types is given, covering all topics of the subject. Ten-point assessment scale: 10 (excellent). Excellent, exceptional knowledge and abilities. 95-100% of questions answered correctly. 9 (very good). Very good knowledge and abilities. 85-94 % of questions answered correctly. 8 (good). Knowledge and abilities are above average. 75-84 % of questions answered correctly. 7 (average). Average knowledge and abilities; there are a few non-essential mistakes. 65-74 % of questions answered correctly. 6 (satisfactory). Knowledge and abilities are below average; there are mistakes. 55-64 % of questions answered correctly. 5 (weak). Knowledge and abilities meet the minimum requirements. 51-54 % of questions answered correctly. 4, 3, 2, 1. Minimum requirements are not met to varying extents.

Artificial Intelligence Policy

Artificial Intelligence (AI) tools are valuable for supporting learning and creativity, but they cannot replace critical thinking, independent work, or academic integrity. Students remain fully responsible for the accuracy, originality, and ethical quality of their work. Any use of AI must respect university rules on academic honesty and be disclosed.

The Guidelines on Artificial Intelligence Usage at Vilnius University apply.
https://www.vu.lt/uploads/EN_Translation_Dirbtinio_intelekto_naudojimo_Vilniaus_universitete_gaires_4cf698a49a.pdf

The Guidelines on the Use of Generative Artificial Intelligence at the Faculty of Philology apply.
https://www.flf.vu.lt/dokumentai/GenAI_guidelines_EN.pdf

Permitted Uses of AI (with disclosure):
Brainstorming, generating ideas, or refining research questions.
Checking grammar, style, and clarity of texts written by the student.
Supporting coding or data handling tasks as long as the student understands, can correct and explain the output.

Prohibited Uses of AI that are considered violations of academic ethics:
Submitting AI-generated content as one's own work (plagiarism).
Using AI to produce subject-specific content for graded assignments intended to assess whether students have acquired and can independently demonstrate the competences developed in the course. That includes generating responses during exams, tests, or quizzes and submitting those answers as their own work.
Relying on AI output without critical evaluation, source checking, or acknowledgment.

Responsibilities:
Students must acknowledge and cite AI use in written assignments when AI has been used (following university citation guidelines).
Students must be able to explain, amend, and justify AI-assisted work.
Violations of this policy may be treated as breaches of academic integrity, with penalties imposed in accordance with university regulations.

Author (-s)	Publishing year	Title	Issue of a periodical or volume of a publication	Publishing house or web link
Required reading				
VU	2025	Virtual learning environment		https://emokymai.vu.lt
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein	2022	*Introduction to Algorithms (4th edition)*	ISBN 0-262-04630-X. 1312 pp (excerpts)	MIT Press and McGraw-Hill. 3rd edition: https://www.cs.mcgill.ca/~akroit/math/compsci/Cormen%20Introduction%20to%20Algorithms.pdf
Steven S. Skiena	2020	*The Algorithm Design Manual (3rd edition)*	ISBN: 9780387948607 (ISBN10: 0387948600), 793 pages (excerpts)	Springer
University of Oslo	2020	*Algorithmic Thinking for the Humanities*	(excerpts)	https://uio-ccse.github.io/algoritmisk-tenkning-humanister/intro.html
Recommended reading				

Updated: 20.10.2026