COURSE UNIT (MODULE) DESCRIPTION

Course unit title	Course unit code
ALGORITHM THEORY AND DATA STRUCTURES	

Lecturer (s)	Department where course unit is delivered
Dr Darius Dilijonas	Vilnius University Kaunas Faculty Institute of Social Sciences and Applied Informatics

Cycle	Level of course unit	Type of the course unit
First	1/1	Mandatory

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Class work, lectures	Autumn semester	English

Prerequisites and corequisites	
Prerequisites: Understanding principles of programming and maths	**Corequisites:**

Number of ECTS credits allocated	Student‘s workload	Contact work hours	Individual work hours
5	130	52	78

Purpose of the course unit: programme competences to be developed

This course provides an introduction to the design and analysis of fundamental data structures and algorithms. Basic data structures to be covered including arrays, lists, stacks, queues, trees, heaps, priority queues, hash tables, and graphs. Algorithms include searching, sorting, tree and graph traversal using the above data structures. Algorithm analysis techniques will be emphasized when using the data structures for designing efficient algorithms.

In this course, several fundamental algorithms and data structures in computer science will be explored. Students will implement some algorithms in C++. Some of the data structures to be encountered include linked lists, stacks, queues, trees, heaps, hash tables, and graphs. Algorithms for searching, traversing trees, hashing, manipulating priority queues, sorting, finding shortest paths in graphs, and much more will be studied and analyzed.

The basic idea of this course is to help students understand many of the fundamental data structures of computer science. With an appreciation for data structures and algorithms and practical experience in implementing them, one can be a much more effective designer, developer, or customer for new applications. Elegant algorithms are also a nice counterpoint to the crafty code and weird features we encounter in daily work.

Learning outcomes of course unit	Teaching and learning methods	Assessment methods
Understand dictionary/search data structures (lists, trees, hash tables).	Lectures, analysis and reading of assigned literature	Exam (to solve problems using analytical methods and programming skills
Understand graph representations and algorithms.		

Understand time and space analysis for both iterative and recursive algorithms and be able to prove the correctness a non-trivial algorithm.		acquired from lectures and assignments to demonstrate knowledge and comprehension of course material.), open questions 3-4, practical cases to solve
Be able to translate high-level, abstract data structure descriptions into concrete code.	LAB/Practical Work; individual work on assigned tasks; practical exercises	Practical assignments to implement data structures and algorithms using a high-level programming language
Understand how real-world problems map to abstract graph problems.		
Be able to communicate clearly and precisely about the correctness and analysis of basic algorithms (both oral and written communication).		

Course content: breakdown of the topics	**Contact work hours**							**Individual work hours and tasks**	
	Lectures	Consultations	Seminars	Practice classes	Laboratory	Practice	**All contact work**	**Individual work**	**Tasks**
Introduction and Short Overview of C++	2				4		6	8	Literature analysis; LAB/practical work; individual work on assigned tasks; practical exercises
Algorithm Analysis	2				4		6	8	Literature analysis; LAB/practical work; individual work on assigned tasks; practical exercises
Arrays, Lists, Stacks and Queues	2				4		6	8	Literature analysis; LAB/practical work; individual work on assigned tasks; practical exercises
Trees and Binary Search Trees	2				4		6	9	Literature analysis; LAB/practical work; individual work on assigned tasks; practical exercises
Search Trees and Multiway-Search Trees	2				4		6	9	Literature analysis; LAB/practical work; individual work on assigned tasks; practical exercises
Hashing and Priority Queues (Heaps)	2				4		6	9	Literature analysis; LAB/practical work; individual work on assigned tasks; practical exercises
Sorting	2				4		6	9	Literature analysis; LAB/practical work; individual work on assigned tasks; practical exercises
Graphs and Advanced Data Structures and Implementation	2				4		6	9	Literature analysis; LAB/practical work; individual work on assigned tasks; practical exercises
Assignments, midterm exams and final exam		4					4	9	Literature analysis, preparation for assignments and exam
Total	16	4	0	0	32	0	52	78	

Assessment strategy	Comparative weight percentage	Date of examination	Assessment criteria
Assignment 1	20%	Week 9	Each assignment may have a mixture of written and programming components. Each assignment will specify the material to be turned in. All programming will be in C/C++. Grading scale (based on 10 points) 10 /A (excellent) Excellent, exceptional knowledge and abilities 9 /A (very good) Strong, good knowledge and skills 8 /B (well) Better than average knowledge and skills 7 /C (moderate) Average knowledge and abilities, there are minor errors 6 /D (satisfactory) Knowledge and skills (skills) are lower than average, there are errors 5 /D (weak) Knowledge and skills (skills) meet the minimum requirements 4, 3, 2, 1 /F (unsatisfactory) Minimum requirements are not met Individual work.
Assignment 1	20%	Week 16	
Midterm Exam 1	10%	Week 7	On exams, students are asked to solve problems using analytical methods and programming skills acquired from lectures and assignments to demonstrate knowledge and comprehension of course material Grading scale (based on 10 points) 10 /A (excellent) Excellent, exceptional knowledge and abilities 9 /A (very good) Strong, good knowledge and skills 8 /B (well) Better than average knowledge and skills 7 /C (moderate) Average knowledge and abilities, there are minor errors 6 /D (satisfactory) Knowledge and skills (skills) are lower than average, there are errors 5 /D (weak) Knowledge and skills (skills) meet the minimum requirements 4, 3, 2, 1 /F (unsatisfactory) Minimum requirements are not met Will be in regular classroom.
Midterm Exam 2	10%	Week 13	
Final exam	40%	January	On exams, students are asked to solve problems using analytical methods and programming skills acquired from lectures and assignments to demonstrate knowledge and comprehension of course material Grading scale (based on 10 points) 10 /A (excellent) Excellent, exceptional knowledge and abilities 9 /A (very good) Strong, good knowledge and skills 8 /B (well) Better than average knowledge and skills 7 /C (moderate) Average knowledge and abilities, there are minor errors 6 /D (satisfactory) Knowledge and skills (skills) are lower than average, there are errors 5 /D (weak) Knowledge and skills (skills) meet the minimum requirements 4, 3, 2, 1 /F (unsatisfactory) Minimum requirements are not met Exam, open questions 3-4 (70% of exam grade), solution of data structure and algorithm 1-2 (30% of exam grade)

Author	**Year**	**Title**	**Number of periodical publication or publication Volume**	**The place of publication and publisher or online link**
Required reading				
Weiss, Mark Allen	2014	Data structures and algorithm analysis in C++	Fourth edition; *ISBN-13: 978-0-13-284737-7*	Pearson Education, Inc.
Robert Lafore, Sams	1999	Teach Yourself Data Structures and Algorithms in 24 Hours Paperback	ISBN-10: 0672316331;	Sams
Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran	1998	Computer Algorithms;	ISBN 0-7167-8316-9;	W. H. Freeman,
Thomas H. Cormen [et al.].	2009	Introduction to algorithms	ISBN 978-0-262-03384-8	Massachusetts Institute of Technology
Recommended reading				
Michael T. Goodrich, Roberto Tamassia, David M. Mount	2011	Data Structures and Algorithms in C++	Second Edition; ISBN-13 978-0-470-38327-8	John Wiley & Sons, Inc.
KRUSE, ROBERT L.	2000	Data structures and program design in C++	ISBN 0–13–087697–6	Prentice-Hall, Inc.