COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Distributed Systems	**DIST7134**

Lecturer(s)	Department where the course unit is delivered
Coordinator: assoc. prof. Karolis Petrauskas **Other lecturers: -**	Department of Software Engineering, Institute of Compute Science, Vilnius University

Cycle	Level of course unit	Type of the course unit
Second	-	Elective

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	Fall semester, second year of study	Lithuanian, English

Prerequisites and corequisites	
Prerequisites: "Software Engineering Models and Methods" or "Modelling and Verification of Software-based Systems".	**Corequisites (if any): -**

Number of ECTS credits allocated	Student's workload	Contact hours	Self-study hours
5	135	52	83

Purpose of the course unit: programme competences to be developed		
To deepen knowledge in design of distributed systems, core algorithms and abstractions in this field, to develop skills to reason about algorithms formally, implement them taking verification into consideration.		
Learning outcomes of the course unit: students will be able to	**Teaching and learning methods**	**Assessment methods**
Design a distributed system by applying core abstractions and algorithms in this field.	Lectures, problem-oriented teaching, case studies, information retrieval, literary reading, individual work, tutorials, laboratory work.	Laboratory works and presentation of their results, written exam (open, semi-open and close-ended questions and tasks).
Formalize a design of a distributed system, formally define specific properties and verify them.		
Analyze scientific papers in the field of distributed systems, understand their contribution in the context of core knowledge.		
Develop fault tolerant systems in Erlang/OTP or Elixir platforms.		

Course content: breakdown of the topics	Contact hours							Self-study work: time and assignments	
	Lectures	Tutorials	Seminars	Practice	Laboratory work	Practical training	**Contact hours**	**Self-study hours**	**Assignments**
1. Introduction, definition of a distributed system, fault modes, main theoretical results (FLP, CAP).	2						**2**	**2**	Self-study of literature.
2. Developing parallel and distributed systems with Erlang/	2				4		**6**	**10**	1st laboratory work: create a simple distributed system (e.g. chat, NTP, etc.)

OTP.									with Erlang/OTP or Elixir, prepare automated distributed tests, a formal specification for it, present results in a class. Self-study of literature.
3. Basic proof techniques, formal specifications and proofs using TLA$^+$.	2						**2**	**6**	Self-study of literature.
4. Communication abstractions: atomic broadcast, read-write registers, distributed state machines. Their proofs.	4				6		**18**	**9**	2nd laboratory work: to develop an application using communication abstractions, distributed clocks, transactions or fault detectors. Prepare automated distributed tests, a formal specification, define its properties formally and present results in a class. Self-study of literature.
5. Clocks in a distributed system, partial and causal order.	2							**6**	
6. Agreement. Distributed transactions, mutual exclusion, deadlock detection.	4							**9**	
7. Fault tolerance: fault detectors, voting.	2							**6**	
8. Fault tolerance: replication (View-stamped replication, Paxos and its variants, Raft and other).	6				6		**16**	**13**	3rd laboratory work: to develop an application involving fault tolerance and replication. Prepare automated distributed tests, a formal specification, define its properties formally, prove one of those properties and present results in a class. Self-study of literature.
9. Fault tolerance: data types (CRDT, MRDT, monotonic data types and other).	2							**6**	
10. Fault tolerance: byzantine faults.	2							**4**	
11. Architecture of distributed databases (Dynamo, Bigtable and other).	2						**2**	**4**	Self-study of literature.
12. Review and analysis of the current research in the topic of distributed systems.	2						**2**	**4**	Self-study of literature.
13. Preparing for the exam and taking the final exam (written)							**4**	**4**	Self-study of literature. Consultation before exam – 2 hours, exam – 2 hours.
Total	**32**				**16**		**52**	**83**	

Assessment strategy	Weight,%	Deadline	Assessment criteria
1st laboratory work	15	4th week of the semester	The assessment of the laboratory works. For each fully completed and timely defended laboratory work, 1 point (of 10) is awarded. If the work is done partially, in poor quality or late, the points are reduced. Lateness no more than 2 weeks leads to reducing the assessment in 25%, lateness no more than 4 weeks – 50%, later – 75%.
2nd laboratory work	15	9th week of the semester	
3rd laboratory work	20	15th week of the semester	
Exam (written)	50	Exam session	Exam consists of open, semi-open and close-ended questions from the topics covered in lectures. The exam is allowed only when all the laboratory works are completed and defended for at least 5 points (of 10, not considering reduction for lateness). At least 50% of the exam points must be collected to pass the exam.

Author	Year	Title	Number or volume	Publisher or URL
Required reading				
Sunil Kumar	2017	Distributed Systems: Design Concepts		Alpha Science Intl Ltd. ISBN: 978-1-84265-933-5
Michael Raynal	2018	Fault-Tolerant Message-Passing Distributed Systems An Algorithmic Approach		Springer Nature Switzerland AG. ISBN: 978-3-319-94140-0
Francesco Cesarini, Steve Vinoski	2016	Designing for Scalability with Erlang/OTP: Implement		O'Reilly Media Inc. ISBN: 978-1449320737

		Robust, Fault-Tolerant Systems		
Recommended reading				
Mikito Takada	–	Distributed Systems for Fun and Profit		Online: http://book.mixu.net/distsys/
Daniel J. Velleman	2006	How To Prove It: A Structured Approach, Second Edition		Cambridge University Press. ISBN: 978-0-512-67599-4
Leslie Lamport	2003	Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers		Addison-Wesley, Pearson Education Inc., Online: https://lamport.azurewebsites.net/tla/book-02-08-08.pdf, ISBN: 0-321-14306-X
David Gries, Fred B. Schneider	1993	A Logical Approach to Discrete Math		Springer-Verlag New York, Inc. ISBN: 0-387-94155-0
Fred Hebert	2013	Learn You Some Erlang for Great Good!		No Starch Press. Online: https://learnyousomeerlang.com/, ISBN: 978-1-593-27435-1