COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Functional Programming	

Lecturer(s)	Unit
Coordinator: Viačeslav Pozdniakov **Other lecturers:**	Department of Software Engineering Institute of Computer Science Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
1st (BA)	Compulsory

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	3 semester	English

Prerequisites
Prerequisites: Procedural Programming, Object Oriented Programming

Number of credits allocated	Student's workload	Contact hours	Individual work
5	130	66	64

Purpose of the course unit: programme competences to be developed

Purpose of the course unit – provide functional programming basics, and introduce modern functional programming languages.

Generic competences:

- An ability to organize their own work independently *(GK1.3)*.
- An ability to undertake literature searches and analysis, and to use databases and other sources of information *(GK2.2)*.
- An ability independently to acquire new knowledge, methodologies, and tools and to apply them in practice *(GK2.3)*.

Specific competences:

- Knowledge and skills of underlying conceptual basis *(SK4)*.
- An ability to design, implement, and evaluate a computer-based system, process, component, or service to meet desired needs *(SK5.3)*.
- An ability to select and use appropriate current techniques, models, solution patterns, skills, and tools necessary for software engineering practice involving emerging application areas *(SK6.2)*.

Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
Understand the principles of functional programming and recognize them.	Lectures, discussions, group project, self-dependent reading.	Written exam, presentation of the group project assignments
Write stateless (without any variables) programs.		
Investigate features of any other functional programming languages.		
Apply functional programming design patterns.		

Course content: breakdown of the topics	Contact hours							Individual work: time and assignments	
	Lectures	Tutorials	Seminars	Practice	Laboratory work (LW)	*Tutorial during LW*	**Contact hours**	**Individual work**	**Assignments**
Functions, types, lists, tuples	2				2		**4**	**3**	
Recursion, tail recursion	2				2	*2*	**4**	**3**	Self-dependent reading. Laboratory work 1.
ADT, classes, instances	2				2		**4**	**3**	
Function composition	2				2		**4**	**3**	
Higher-order functions	2				2		**4**	**3**	
Monads, do-notation	4				4	*2*	**8**	**7**	Self-dependent reading. Laboratory work 2.
Functors, Applicative functors	2				2		**4**	**3**	
Free Monads	4				4		**8**	**7**	
Reader, Writer, State monados	2				2	*2*	**4**	**3**	Self-dependent reading. Laboratory work 3.
STM	2				2		**4**	**3**	
Monad transformers	2				2		**4**	**3**	
Monoids, Traversables, Foldables	2				2	*2*	**4**	**3**	Self-dependent reading. Laboratory work 4.
Lazy evaluation, exceptions.	2				2		**4**	**3**	
Tagless-Final style	2				2		**4**	**3**	
Preparation for exam, exam itself		1					**2**	**14**	1 h for tutorial 1 h for the exam 13 h for preparation
Total	**32**	**1**			**32**	*8*	**66**	**64**	

Assessment strategy	Weight %	Deadline	Assessment criteria
Exam	60%	January	All correctly answered exam tasks give 6 points. A student can take part in the examination only if they have collected at least 1 point for laboratory works.
Laboratory work 1	10%	Week 4	Correctly written program gives max 1 point. One week penalty after deadline – 0.1 points.
Laboratory work 2	10%	Week 8	
Laboratory work 3	10%	Week 12	
Laboratory work 4	10%	Week 16	

Author	Publishing year	Title	Number or volume	Publisher or URL
Required reading				
Graham Hutton	2016	Programming in Haskell, 2nd edition		Cambridge University Press
Alexander Granin	2023	Functional Design and Architecture		Manning, MEAP
Recommended reading				
Bryan O'Sullivan, John Goerzen, and Don Stewart	2009	Real World Haskell		O'Reilly
Miran Lipovača	2011	Learn You a Haskell for Great Good!		http:// learnyouahaskell.com