



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
1 st (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, introduce modern functional programming languages.		
Generic competences: <ul style="list-style-type: none"> • Communication and collaboration (GK1). • Life-long learning (GK2). • Social responsibility (GK3). 		
Specific competences: <ul style="list-style-type: none"> • Knowledge and skills of underlying conceptual basis (SK4). • Software development knowledge and skills (SK5). • Technological and methodological knowledge and skills, professional competence (SK6). 		
Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
Understand principles of functional programming and recognize them. Write stateless (without any variables) programs. Investigate features of any other functional programming languages. Apply functional programming design patterns.	Lectures, discussions, group project, self-dependent reading.	Written exam, presentation of the group project assignments

Course content: breakdown of the topics	Contact hours						Individual work: time and assignments		
	Lectures	Tutorials	Conversations	Presentations	Laboratory work (LW)	Tutorial during LW	Contact hours	Individual work	Assignments
Functions, types, lists, tuples, polymorphic types, overloaded types.	2				2		4	3	
Classes, pattern matching, lambda expressions, conditions, guards.	2				2	2	4	3	
List comprehensions.	2				2		4	3	
Recursion.	2				2		4	3	Self-dependent reading. Group project assignment 2
Higher-order functions	2				2		4	3	
Algebraic data types. Instances.	2				2	2	4	3	Self-dependent reading. Group project assignment 3
Untyped lambda calculus.	4				4		8	7	
Monads	2				2		4	3	Self-dependent reading. Group project assignment 4
Functors, Applicative functors	2				2	2	4	3	
Monoids, Traversable, Foldables	2				2		4	3	
Lazy evaluation, exceptions.	2				2		4	3	Self-dependent reading. Group project assignment 4
Monad transformers	2				2		4	3	
Tagless-Final style	2				2	2	4	3	Self-dependent reading.
Typed lambda calculus	4				4		8	7	
Preparation for exam, exam itself		1					2	14	1 h for tutorial 1 h for exam 13 h for preparation
Total	32	1			32	8	66	64	

Assessment strategy	Weight %	Deadline	Assessment criteria
Exam	50%	January	All correctly answered exam tasks give 5 points. A student can take part in the examination only if he/she has collected at least 1 point for group project.
Group project assignment 1	15%	Week 4	Correctly written program gives 1 points. One week penalty after deadline – 0.1 points.
Group project assignment 2	15%	Week 8	
Group project assignment 3	15%	Week 12	Correctly written program gives 2 points. One week penalty after deadline – 0.2 points.
Group project assignment 4	15%	Week 16	Correctly written program gives 1 points. One week penalty after deadline – 0.1 points.

Author	Publishing year	Title	Number or volume	Publisher or URL
Required reading				
Graham Hutton	2016	Programming in Haskell, 2 nd edition		Cambridge University Press
Benjamin C. Pierce	2002	Types and Programming Languages		The MIT Press
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