COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Python Programming	**ITPY**

Lecturer(s)	Department where the course unit is delivered
Coordinator: Tomas Raila	Department of Computational and Data Modeling Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
First	Optional

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	4th semester	Lithuanian and English

Prerequisites
Student should have completed the following (or equivalent) courses: object-oriented programming (ITOP), data structures (ITDS), database management systems (ITRDB).

Number of ECTS credits allocated	Student's workload	Contact hours	Individual work
5	134	64	70

Purpose of the course unit: programme competences to be developed

Generic competences to be developed

- Ability for abstract thinking , processing and analysing information (BK3)
- Ability to use information and communications technologies (BK5)

Subject-specific competences to be developed

- Ability to apply general methods of the program design, make and analyse software requirements (DK1)
- Ability to analyse the algorithmic process of the task based on the general properties of the algorithm (DK2)
- Ability to do program and IT service testing and debugging (DK4)
- Ability to build conceptual and physical data models based on information management and data modelling principles (DK9)

Learning outcomes of the course unit	Teaching and learning methods	Assessment methods
Ability to write high quality procedural, functional and object-oriented Python code.	Lectures, live programming, study of literature and code examples	Practical tasks and their presentations, exam.
Ability to effectively develop, analyze, test and debug Python programs by using apropriate tools.	Lectures, live programming, study of literature and code examples	
Knowledge of most widely used Python libraries and ability to select and apply them in specific tasks.	Lectures, live programming, study of library documentation and code examples	
Understanding of Python interpreter, ability to manage Python installations and packages.	Lectures, demos, study of literature	

Course content: breakdown of the topics	Individual work: time and assignments							
	Lectures	**Tutorials**	**Seminars**	**Laboratory work**	**Consulting during lab. work**	**Contact hours**	**Individual work**	**Assignments**
1. Course introduction. Overview of Python language.	2			2		**4**	**2**	
2. Basic data structures and control flow, syntax rules.	3			3	6	**6**	**6**	Individual study of course material, practical exercises, homework.
3. Functions, iterators and generators.	3			3		**6**	**6**	
4. Exception handling and debugging. Basics of unit testing.	3			3		**6**	**6**	
5. File and I/O operations, serialization.	2			2		**4**	**4**	
6. Classes and objects, special methods.	3			3		**6**	**6**	
7. Modules, packages and environments. Python interpreter.	2			2		**4**	**4**	
8. Standard Python library.	2			2		**4**	**4**	
9. Concurrency. Multithreading and multiprocessing.	2			2		**4**	**4**	
10. Network programming, sockets, asyncio.	2			2		**4**	**4**	
11. Web programming with Flask framework.	3			3		**6**	**6**	
12. Working with relational databases, ORM libraries.	2			2		**4**	**4**	
13. Basics of data analysis. Numpy, scipy, pandas, matplotlib libraries.	3			3		**6**	**6**	
14. Preparation for exam							**8**	
Total:	**32**			**32**		**64**	**70**	

Assessment strategy	Weight %	Deadline	Assessment criteria
Practical tasks	50	Custom, defined	Four programming tasks of varying complexity, implemented during the course. Each completed task has to be individually presented to lecturer during exercise sessions, before scheduled deadline. Scores are assigned according to fulfillment of task requirements, correct working of implemented code and student's ability to explain it.
Exam	50	End of semester	Written exam consisting of 10-20 questions of various types (open-ended and closed-ended).

Author	Publishing year	Title	Issue No or volume	Publishing house or Internet site
Required reading				
Python Software Foundation	2019	The Python Tutorial		https://docs.python.org/3/tutorial
Mark Pilgrim	2010	Dive Into Python 3		APress
Optional reading				
Allen B. Downey	2012	Think Python. How to Think Like a Computer Scientist		https://greenteapress.com/wp/think-python-2e/
David Beazley, Brian K. Jones	2013	Python Cookbook		O'Reilly