COURSE UNIT (MODULE) DESCRIPTION

Course unit (module) title	Code
Python for Data Science	

Lecturer(s)	Department(s) where the course unit (module) is delivered
Coordinator: Martynas Arbačiauskas - 32 h lectures, 32 h practicals	Life Science Center, Institute of Biosciences, Vilnius University

Study cycle	Type of the course unit (module)
Master Studies	Elective

Mode of delivery	Period when the course unit (module) is delivered	Language(s) of instruction
Lectures and Exercises	Autumn semester	Lithuanian/English

Requirements for students	
Prerequisites: Basic computer literacy	**Additional requirements (if any):** Equipment: personal computer

Course (module) volume in credits	Total student's workload	Contact hours	Self-study hours
5	128	64	64

Purpose of the course unit (module): programme competences to be developed		
Due to an increasing amount of data worldwide there is a growing demand for related skillsets and tools. Python is a programming language that has become especially popular for use in scientific fields such as Data Science and Machine Learning. Its simple syntax makes it a great choice for beginner programmers. This course familiarizes students with basic principles of programming for Data Science. Students will learn to operate Python objects, write their own functions, and apply them to biological data. Upon completion, students will know how to use Python to process, analyze, and visualize real-world biological datasets for data-driven research.		
Learning outcomes of the course unit (module)	**Teaching and learning methods**	**Assessment methods**
Upon course completion, students should be able to design, code, test, and debug Python language programs, as well as use Python libraries to process, analyze, and visualize data.	Lectures, code demonstration and discussion, homework assignments, live coding sessions.	Students will complete graded homework assignments that require writing code. Midterm and final exams will consist of live coding sessions where students demonstrate their programming skills in real-time, as well as explain their work in oral examination.

Content: breakdown of the topics	Contact hours						Self-study work: time and assignments	
	Lectures	Tutorials	Seminars	Exercises	Laboratory work	**Contact hours**	**Self-study hours**	**Assignments**
Course Introduction and Python Basics	2			2		4	4	Homework 1
String and List Variables	2			2		4	4	Homework 2
Control Flow: Conditions and Loops	2			2		4	4	Homework 3
Writing Custom Functions	2			2		4	4	Homework 4
Advanced Collections and File Parsing	2			2		4	4	Homework 5
IDEs, Virtual Environments and Git	2			2		4	4	Homework 6
Introduction to Numpy: Arrays and Slicing	2			2		4	4	Homework 7
Introduction to Pandas: Dataframes and Quick Summaries	2			2		4	4	Homework 8
Data Cleaning	2			2		4	4	**Midterm**
Data Transformation: Joining and Reshaping	2			2		4	4	Homework 9
Data Visualization	2			2		4	4	Homework 10
Statistics in Python	2			2		4	4	Homework 11
Introduction to Machine Learning and Data Preprocessing	2			2		4	4	Homework 12
Supervised Learning: Regression and Classification	2			2		4	4	Homework 13
Unsupervised Learning and Model Evaluation	2			2		4	4	Homework 14
Course Summary	2			2		4	4	
Total	**32**			**32**		**64**	**64**	

Assessment strategy	Weight, %	Deadline	Assessment criteria
Homework	20	During semester	Short weekly homework exercises. 12 out of 14 should be submitted on time to achieve the maximum grade. Assessment criteria: ● An expected input and output are provided in the task, and are achieved.
Midterm assessment	30	In the middle of semester	An open-book live coding session, during which students should complete a given assignment by writing code. Assessment criteria: ● 40% – the solution provides the expected output ● 30% – the code is easy to read (style is consistent, variables are named meaningfully), does not contain irrelevant parts ● 30% – the student is able to clearly explain their design decisions and functions used (3 oral questions of equal weight)
Final assessment	50	At the end of semester	An open-book live coding session, during which students should complete a given assignment by writing code. Assessment criteria are the same as for the midterm assessment.
Optional assignments	Up to 20%	During semester	Optional bonus assignments to be agreed upon in advance with the lecturer: - Students can be graded on their own programming project (up to 10% of the final grade depending on the scope) - Students can prepare a 20-minute presentation on a Python package (up to 10% of the final grade). It should include a brief demonstration and be relevant to biological data analysis.

Author	Year of publication	Title	Issue of a periodical or volume of a publication	Publishing place and house or web link
Compulsory reading				
Wes McKinney	2022	Python for Data Analysis	3rd Edition	O'Reilly Media https://wesmckinney.com/book/
Allen B. Downey	2024	Think Python	3rd Edition	O'Reilly Media https://allendowney.github.io/ThinkPython/
Jake VanderPlas	2016	Python Data Science Handbook	1st Edition	O'Reilly Media https://jakevdp.github.io/PythonDataScienceHandbook/
Additional reading				
Claus O. Wilke	2019	Fundamentals of Data Visualization	1st Edition	O'Reilly Media https://clauswilke.com/dataviz/
Russell A. Poldrack	2019	Statistical Thinking for the 21st Century	2024 revision	https://statsthinking21.github.io/statsthinking21-core-site/