



COURSE UNIT (MODULE) DESCRIPTION

Course unit (module) title	Code
Programming „Python“	

Academic staff	Core academic unit(s)
Coordinating: Assist. dr. Konstantinas Korovkinas	Kaunas faculty Institute of applied Informatics Muitines str. 8, LT-44280 Kaunas

Study cycle	Subject (module) level	Type of the course unit
First	1/1	Optional

Mode of delivery	Semester or period when it is delivered	Language of instruction
Auditorium	4 semester	English

Requirements for the student	
Prerequisites: Structured and object-oriented programming	Adjacent requirements:

Number of ECTS credits allocated	Student's workload (total)	Contact Hours	Individual work
5	130	52	78

Purpose of the course unit		
Subject goal – to introduce the possibilities of programming in Python and to provide practical skills.		
General competence: 1. Continuous learning (BK2)		
Subject competencies: 1. Knowledge and skills of conceptual basics (DK4)		
2. Technological, methodological knowledge and skills, professional competence (DK6)		
Learning outcomes of the course unit	Teaching and learning methods	Assessment methods
Will be able to describe Python programming language syntax and principles.	Lectures, preparation of practical tasks	Practical tasks, exam
Will be able to understand a source code written in Python, to modify and execute it.		
Will be able to write applications in Python.		
Will be able to work with text files.		
Will be able to work with databases.		
Will be able to develop web services.		

Content	Contact Hours							Individual work: time and assignments	
	Lectures	Tutorials	Seminars	Workshops	Laboratory work	Internship	Contact hours, total	Individual work	Tasks for individual work
1. Python interpreter, programming language syntax (procedural programming, object-oriented programming) and style	1			3			6	2	Program writing (independent program preparation). Analysis of literature and case studies.
2. Modules and standard library	1			3			6	6	Program writing (independent program preparation). Analysis of literature and case studies.
3. Work with text files	3			4			6	6	Program writing (independent program preparation). Analysis of literature and case studies. Defense of the first practical work.
4. Python object-oriented programming (OOP)	3			6			10	16	Program writing (independent program preparation). Analysis of literature and case studies. Defense of the second practical work.
5. Work with databases	3			6			10	16	Program writing (independent program preparation). Analysis of literature and case studies. Defense of the third practical work.
6. Web services creation	3			6			6	8	Analysis of literature and case studies.
7. Popular Python Libraries	2			4			6	8	Analysis of literature and case studies.
8. Preparation for the exam		4						16	
Exam							2		
Total	16	4		32			52	78	

Assessment strategy	Weight %	Deadline	Assessment criteria
Practical tasks (P)	50%	During semester	Three practical assignments (P1, P2, P3). Each practical assignment is graded on a scale of 10 points. Assessment criteria: • correct operation and fulfillment of the specified conditions (20%), • error handling (20%), • report (10%), • program explanation (50%). The overall grade for the practical assignments is calculated using the formula: $P = (P1+P2+P3)/3$

Exam(E)	50%	During exam session	The exam consists of three programming tasks. The maximum score is 10 points. Evaluation criteria: • Correct functionality and fulfillment of the specified conditions (50%), • Error handling (30%), • Commented code (20%).
Final mark (FM)	100%	During exam session	Ten-point proportional knowledge assessment system is used. Exam is passed if final mark (FM) is ≥ 5 . Formula of the final mark (FM): $FM = 0,5 * P + 0,5 * E$, where P – the overall grade for the practical assignments, E – exam mark,
Extern exam assessment strategy			Not applicable
Provisions for the use of artificial intelligence			AI can be used to edit course practical work. AI is not allowed to be used during the defence of the practical tasks and exam.

A student who (1) throughout the semester consistently fails to demonstrate progress in achieving the expected learning outcomes of a subject (module) during the practical classes (seminars, exercises, laboratory work, etc.) and (2) fails to complete all interim assessment requirements and tasks within the time specified in the course description, is not allowed to participate in the examination session.

Author (-s)	Publishing year	Title	Issue of a periodical or volume of a publication	Publishing house or web link
Required reading				
James P. Meyers	2023	Python Programming Bible		https://www.amazon.com/Python-Programming-Bible-Including-Practical/dp/B0CL2VCGYM
David Beazley	2021	Python Distilled		Pearson
Recommended reading				
Python documentation.		The Python documentation		https://docs.python.org/3/
Sqlalchemy documentation		Sqlalchemy documentation		https://www.sqlalchemy.org/
Flask documentation		Flask documentation		https://flask.palletsprojects.com/en/3.0.x/