MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
MEĐIMURJE UNIVERSITY OF APPLIED SCIENCES IN ČAKOVEC

COURSE SYLLABUS

ACADEMIC YEAR: 2025/2026

1. GENERAL COURSE INFORMATION				
1.1 Course name	**Python programming**			
1.2 Study program/s	Undergraduate Professional Study Programme Computer Engineering			
1.3 Course status (O,E)	E	**1.6 Mode of instruction (number of hours)**	**Lectures**	30
1.4 Course code			**Exercises**	30
1.5 Course abbreviation			**Seminars**	
1.6 Semester	V		**E-learning**	Merlin
1.7 ECTS	5	**1.7 Place and time of instruction**	The premises of the Međimurje University of Applied Sciences in Čakovec, according to the schedule published on the website	
2. TEACHING STAF.F				
2.1 Course leader/s-title	PhD, Sanja Obradović, prof.	**contact**	sanja.obradovic@mev.hr	
		contact		
2.2 Assistant/s- title		**contact**		
		contact		
2.3 Instruction held by- title	PhD, Sanja Obradović, prof.	**contact**	sanja.obradovic@mev.hr	
2.4 Course leader/s-title		**contact**		
3. COURSE DESCRIPTION				
3.1. Course goals	The aim of the course is to equip students with the skills to independently solve practical programming tasks using the Python programming language and to develop applications based on modern technologies and tools. Throughout the course, students will master both fundamental and advanced programming concepts, ranging from basic data structures to object-oriented programming, working with databases, web development frameworks, and task automation. Students will gain the ability to build real-world applications, develop their own portfolios, and be prepared to apply the acquired knowledge in the job market, including positions related to software development, web applications, and process automation. Special emphasis is placed on the practical application of knowledge through project development, working with repositories, and a collaborative team environment.			
3.2 Prerequisites	None			
3.3 Course outcomes	After successfully completing the course, students will be able to: LO1 – Independently develop and test simple applications in Python using object-oriented programming (OOP) LO2 – Automate everyday tasks and processes using Python scripts, and work with files and API interfaces LO3 – Design and develop web applications using modern Python frameworks LO4 – Work with relational databases using Python to build CRUD applications LO5 – Manage source code using version control systems and document projects			

3.4 Contribution of the course to the study program

SO7 Develop program code in multiple programming languages using modern methods and tools
SO13 Develop applications using the object-oriented paradigm in solving programming tasks
SO16 Develop web and mobile projects, applying advanced technologies, and connecting with databases using modern methods and tools
SO17 Choose the appropriate programming language and technology in solving programming tasks

3.5 Course content

The course covers both fundamental and advanced programming concepts using Python as the primary programming language. Throughout the course, students acquire the knowledge and skills necessary for developing software solutions and applications in real-world environments. Students will master basic concepts such as variables, loops, functions, and data structures, and will build on this knowledge through the application of object-oriented programming (OOP).
Special emphasis is placed on task automation, working with API interfaces, processing data from various sources, and developing web applications using modern Python frameworks. The course also includes working with relational databases and practical exercises that prepare students for developing CRUD applications and collaborating in team environments using version control systems.

3.6 Types of coursework

x	Lectures	X	Exercises		Blended e-learning	X	Individual activities		Laboratory
	Seminars and workshops	x	Distant learning		Field classes	x	Multimedia and network		Mentorship
	Other								

3.7 Language of instruction

Croatian

3.8 Monitoring students' work (enter the number of ECTS credits for each activity so that the total number of ECTS credits is equal to the total ECTS value of the course, 1 ECTS = 30 hours)

2	Class attendance		Seminars		Research
0,5	Midterm exams/written exam		Project		Essay
	Oral exam	1,5	Practical task		

Activity specification	Percent %	Points
Assessment during instruction		
Tasks during class	10%	10
Practical task	50%	50
Midterm exam 1	20%	20
Midterm exam 2	20%	20
Exam assessment for the students who failed to fulfil all the obligatory requirements during the semester		
Written exam	*40%*	*40*
Total:	**100%**	**100**

Ways of evaluating learning outcomes					
	Tasks in class	**Mid-term exam 1**	**Mid-term exam 2**	**Practical work**	**Total**
Outcome 1	5	20			25
Outcome 2	5		20		25
Outcome 3				30	30
Outcome 4				10	10
Outcome 5				10	10
Total	**10**	20	20	50	100

Grading of outcomes (in order to pass the mid-term exam/exam the student must achieve at least 50% points for each learning outcome)
Points Grade

	89 – 100 excellent (5) *76 – 88 very good (4)* *63 – 75 good (3)* *50 – 62 pass (2)* *0 – 49 fail (1)*
3.9 Specific features related to taking the course	If a student collects 50% of the points of each outcome, he / she directly takes the exam, if he / she has submitted a practical task. A student cannot take the exam if he / she has not submitted a practical task. The practical task is made according to the instructions published on the Merlin e-learning system and is submitted by placing it on the Merlin. The practical task is submitted at least 3 days before the exam. During the exam, it is possible to verbally check the knowledge in the preparation of practical task. If a student does not achieve enough points on the midterm exam, he / she cannot take the next midterm exam. Once achieved points in intermediate exams for each learning outcome are no longer deleted unless the student decides to correct the result for each learning outcome, whereby the points won until then are deleted and newly achieved points for that learning outcome are entered.
3.10 Students obligations	• a full-time student has the right to sit for the exam if he attends classes for a minimum of 70% of the total prescribed number of hours • a full-time student who attends classes from 50 to 70% of the total prescribed number of hours can exercise the right to take the exam by completing additional teaching activities in agreement with the course teacher • a full-time student who attends a certain course for less than 50% of the prescribed number of hours enrols in the course the following academic year • a part-time student has the right to sit for the exam if he/she attends classes for a minimum of 30% of the total prescribed number of hours • a part-time student who attends classes for 20 to 30% of the total prescribed number of hours can exercise the right to sit for the exam by completing additional teaching activities in agreement with the course teacher • a part-time student who attends the classes of a certain course for less than 20% of the prescribed number of hours re-enrols in the course the following academic year
3.11 Written assignments	
3.12 Required reading	1. Eric Matthes, "Python Crash Course: A Hands-On, Project-Based Introduction to Programming", 3rd Edition, No Starch Press, 2023.
3.13 Additional reading	
4. ADDITIONAL INFORMATION ABOUT THE COURSE	
4.1 Quality control	The quality of the program, teaching process, teaching skills and level of mastery of the material will be established by conducting a written evaluation based on questionnaires, and in other standardised ways and in accordance with the by-laws of the Međimurje University of Applied Sciences in Čakovec.
4.2 Contact the teacher	Students can contact the teacher during the office hours and during classes. All other methods of communication are arranged with the teacher. It is also possible to ask questions by e-mail, which will be answered in 48 hours at the latest. It is desirable for students to come as often as possible for any possible questions during the teacher's office hours.

4.3 Information about the course	It is the obligation of each student to be regularly informed about the course. All notifications about the classes or possible postponement of classes will be posted on the bulletin board and on the website of the University at least 24 hours in advance.

5. ELABORATION OF THEMATIC UNITS

Week	Topic	Course outcome
1	Introduction to the course, basic programming concepts, Python environment setup	LO1
2	Variables, data types, conditional statements and loops in Python	LO1
3	Functions and working with files	LO1
4	Introduction to OOP – classes and objects	LO1
5	Advanced OOP concepts: inheritance and encapsulation	LO1
6	Task automation and working with files (CSV, JSON)	LO2
7	Working with APIs and retrieving data from external sources	LO2
8	Midterm exam 1	
9	Introduction to web development – basics of working with a modern Python framework	LO3
10	Development of a simple web application (routes, templates, forms)	LO3
11	Working with databases using Python – basics of relational databases	LO4
12	Building a CRUD application connected to a database	LO4
13	Version control systems and teamwork (basics of working with Git and repositories)	LO5
14	Practical tasks: documenting and presenting own projects	LO5
15	Midterm exam 2	