Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

COURSE SPECIFICATION

Course code	full-time programme:	**M#2-S2-ME-EM-112**
	part-time programme:	
Course title in Polish	**Zaawansowane techniki programowania**	
Course title in English	**Advanced Programming Methods**	
Valid from (academic year)	**2024/2025**	

GENERAL INFORMATION

Programme of study	**MECHANICAL ENGINEERING**
Level of qualification	**second-cycle**
Type of education	**academic**
Mode of study	**full-time programme**
Specialism	**Machine Operation and Maintenance**
Department responsible	**Department of Mechatronics and Weapons Engineering**
Course leader	**dr hab. inż. Izabela Krzysztofik, prof. PŚk**
Approved by	**dr hab. Jakub Takosoglu, prof. PŚk, Dean of the Faculty of Mechatronics and Mechanical Engineering**

COURSE OVERVIEW

Course type		**specialism-related**
Course status		**compulsory**
Language of instruction		**English**
Semester of delivery	full-time programme	**Semester I**
	part-time programme	**Semester I**
Pre-requisites		
Examination required (YES/NO)		**NO**
ECTS value		**2**

Mode of instruction		**lecture**	**class**	**laboratory**	**project**	**seminar**
No. of hours per semester	full-time programme	**15**		**15**		
	part-time programme	**9**		**9**		

LEARNING OUTCOMES

Category of outcome	Outcome code	Course learning outcomes	Corresponding programme outcome code
Knowledge	W01	The student has a structured in-depth knowledge of the use of modern programming techniques. Understands the complexity of modern programming languages, as well as the applicability in the context of advanced programs used in mechanical and mechanical engineering issues.	MiBM2_W03
Skills	U01	The student is able to use modern programming techniques to integrate with engineering software. The student is able to prepare own programs for engineering calculations and automation of design preparation tasks. The student is able to integrate own software with commonly used programs by writing execution scripts.	MiBM2_U01 MiBM2_U02
	U02	The student is able to plan the result of work during the process of writing his own software. Implements advanced programming techniques consciously. While performing software integration tasks, he recognizes the need for further self-development in the implementation of advanced programming techniques.	MiBM2_U16
Competence	K01	The student is prepared to critically evaluate his knowledge and the possibility of extending and supplementing his knowledge in the field of programming techniques in the field of mechanics and mechanical engineering.	MiBM2_K01

COURSE CONTENT

Mode of instruction	Topics covered
lecture	Theoretical basics of how high-level programming languages work. Discussion of data structures. Presentation of algorithmics and issues related to the construction of machines that can be implemented using written programs. Introduction to programming techniques that allow communication with programs dedicated to the design and operation of machines. Discussion of Rest APIs and the possibility of automating tasks such as generating documentation or a bill of materials using written programs.
laboratory	Introduction to advanced integrated text editors (IDEs) used in modern programming. Installation of the working environment. Introduction to advanced programming methods and familiarisation with their paradigms. Familiarisation with data structures and the process of interpreting and compiling programs. Introduction to selected design patterns and advanced techniques for creating readable and reusable programming code. Writing programmes for engineering tasks involving data processing and its integration with commonly used software.

ASSESSMENT METHODS

Outcome code	Methods of assessment					
	Oral examination	Written examination	Test	Project	Report	Other
W01			X			
U01			X			X
U02			X			X
K01						X

ASSESSMENT TYPE AND CRITERIA

Mode of instruction	Assessment type	Assessment criteria
lecture	non-examination assessment	Positive pass in the final test. At least 50% of points.
laboratory	non-examination assessment	Positive pass in the partial tests and work during laboratory. The final grade is an arithmetic mean.

OVERALL STUDENT WORKLOAD

ECTS weighting												
No.	Activity type	Student workload										Unit
		full-time programme					part-time programme					
		L	C	Lb	P	S	L	C	Lb	P	S	
1.	Scheduled contact hours	15		15								h
2.	Other contact hours (office hours, examination)	2		2								h
3.	**Total number of contact hours**	**34**										h
4.	**Number of ECTS credits for contact hours**	**1,4**										ECTS
5.	**Number of independent study hours**	**16**										h
6.	**Number of ECTS credits for independent study hours**	**0,6**										ECTS
7.	**Number of practical hours**	**25**										h
8.	**Number of ECTS credits for practical hours**	**1,0**										ECTS
9.	**Total study time**	**50**										h
10.	**ECTS credits for the course** *1 ECTS credit = 25-30 hours of study time*	**2**										ECTS

READING LIST

1. Klabnik Steve, Nichols Carol: The Rust Programming Language. No Starch Press, San Francisco 2018
2. Downey Allen, Elkner Jefrey, Meyers Chris: Think Python. How to Think Like a Computer Scientist. GreenTea Press Wellesley, Massachusetts 2002 (http://www.greenteapress.com/thinkpython/)
3. Pilgrim Martin: Dive into Python 3. (http://www.diveintopython.net/)
4. Wentworth Peter, Elkner Jeffrey, Downey Allen, Meyers Chris: How to Think Like a Computer Scientist: Learning with Python 3. Documentation Release 3rd Edition

5. Bjarne Stroustrup: The C++ Programming Language. Pearson Education, Boston 2013