University of Gdańsk

Subject card

Subject name and code	Introduction to Python programming, PG_00054863		
Field of study	Chemistry		
Date of commencement of studies	October 2024	Academic year of realisation of subject	2024/2025
Education level	postgraduate studies	Subject group	Obligatory subject group in the field of study
Mode of study	full-time studies	Mode of delivery	at the university
Year of study	1	Language of instruction	English
Semester of study	1	ECTS credits	2.0
Learning profile	academic	Assessment form	
Conducting unit	Faculty of Chemistry		
Name and surname of lecturer (lecturers)	Subject supervisor	prof. dr hab. Cezary Czaplewski	
	Teachers	prof. dr hab. Cezary Czaplewski	

Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	0.0	0.0	0.0	15
	E-learning hours included: 0.0						

Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
	Number of study hours	15	2.0	33.0	50
Subject objectives	Ability to properly design basic algorithms, introduction to programming in Python				

<table>
<tr><td>Learning outcomes</td><td>
<table>
<tr><th>Course outcome</th><th>Subject outcome</th><th>Method of verification</th></tr>
<tr><td>[CHEMMU2_W05] Has extended knowledge in the field of the specialisation studied.</td><td>Names and describes the types and the data structures of the Python programming language. Differentiates the Python control instructions.</td><td>[SW4] test/exam - oral or written
[SW2] presentation/project/paper/ report</td></tr>
<tr><td>[CHEMMU2_K06] Undertakes research tasks consciously and responsibly, understanding the social aspects of the practical application of the acquired knowledge and skills and the responsibility related to it.</td><td>Learns the principles of working safely, responsibly, and efficiently using the
workstations connected to the Internet. Develops the responsibility for his/her personal account on the workstation.</td><td>[SK8] observation of student's independent or team work</td></tr>
<tr><td>[CHEMMU2_U11] Communicates in English in accordance with the requirements specified for level B2 of the Common European Framework of Reference for Languages.</td><td>A student can use specialist terminology associated with algorithms and the Python programming language.</td><td>[SU2] presentation/project/paper/ report
[SU5] implementation of a problem task</td></tr>
<tr><td>[CHEMMU2_K01] Knows the limitations of her/his own knowledge; understands the need for further education and can inspire other people to do so.</td><td>Develops the ability of working in a team.</td><td>[SK5] implementation of a problem task
[SK8] observation of student's independent or team work</td></tr>
<tr><td>[CHEMMU2_W06] Applies mathematics to the extent necessary to understand, describe and model chemical processes of medium complexity.</td><td>The student defines the concept of the algorithm. Characterizes the most important classes of algorithms.</td><td>[SW4] test/exam - oral or written
[SW2] presentation/project/paper/ report</td></tr>
<tr><td>[CHEMMU2_U02] Critically assesses the results of conducted, performed observations and theoretical calculations and discusses errors.</td><td>The student designs simple algorithms, writes them using the Python programming
language and then tests the obtained programs.</td><td>[SU2] presentation/project/paper/ report
[SU4] test/exam - oral or written</td></tr>
</table>
</td></tr>
<tr><td>Subject contents</td><td>Algorithms and data structures in Python. Review of the most important data structures that are helpful in programming, objects, expressions, and numerical types. Functions and scoping. Testing and debugging. Handling exceptions. Classes and object-oriented programming. The most important programming algorithms: approximate estimation of the complexity of algorithms, brute-force algorithms, divide and conquer algorithms, recursion. Versions of Python language, useful Python libraries: NumPy, Scipy.</td></tr>
<tr><td>Prerequisites and co-requisites</td><td></td></tr>
<tr><td>Assessment methods and criteria</td><td>
<table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>test exam</td><td>51.0%</td><td>100.0%</td></tr>
</table>
</td></tr>
<tr><td>Recommended reading</td><td>
<table>
<tr><td>Basic literature</td><td>Mark Lutz, Learning Python, O'Reilly Media, Inc, USA, 2013

John V. Guttag Introduction to Computation and Programming Using Python, MIT Press 2013</td></tr>
<tr><td>Supplementary literature</td><td>Alberto Boschetti, Luca Massaron, Python Data Science Essentials, PublishDrive 2018

Svein Linge, Hans Petter Langtangen Programming for Computations Python, Springer 2020

Joakim Sundnes Introduction to Scientific Programming with Python, Springer 2020</td></tr>
<tr><td>eResources addresses</td><td>Adresy na platformie eNauczanie:</td></tr>
</table>
</td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td></td></tr>
<tr><td>Work placement</td><td>Not applicable</td></tr>
</table>

Document generated electronically. Does not require a seal or signature.