KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Course title

Introduction to Python programming

ECTS code

13.3.1295

Name of unit administrating study

null

Studies

faculty	field of study		
Wydział Chemii	Chemia	type	drugiego stopnia
		form	stacjonarne
		specialty	Digital Chemistry
		specialization	wszystkie

Teaching staff

prof. dr hab. Cezary Czaplewski, profesor uczelni; dr hab. Artur Giełdoń; dr hab. Adam Sieradzan, profesor uczelni

Forms of classes, the realization and number of hours

Forms of classes

Laboratory classes, Lecture

The realization of activities

classroom instruction

Number of hours

Lecture: 15 hours, Laboratory classes: 45 hours

ECTS credits

5

Lecture – 15 h

Laboratory classes – 45 h

student’s own work – 30 h

tutorial classes – 35 h

TOTAL: 125 h – 5 ECTS

The academic cycle

2023/2024 winter semester

Type of course

obligatory

Language of instruction

English

Teaching methods

- Case studies in computer laboratory
- multimedia-based lecture

Form and method of assessment and basic criteria for eveluation or examination requirements

Final evaluation

- Graded credit
- Examination

Assessment methods

Lecture – exam with multiple-choice questions

Laboratory classes – written reports

The basic criteria for evaluation

according to “Rules and regulations for studies at the University of Gdansk”

Lectures: passing the final exam in the form of a multiple-choice question test (a score of 50% or more required to pass the exam).

Lab classes: the arithmetic mean of partial grades received during the semester for written reports on laboratory exercises

Method of verifying required learning outcomes

Required courses and introductory requirements

A. Formal requirements

None

B. Prerequisites

None

Aims of education

Ability to properly design basic algorithms, introduction to programming in Python

Course contents

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.

Bibliography of literature

Literature required to pass the course

John V. Guttag Introduction to Computation and Programming Using Python, MIT Press 2013

Extracurricular readings

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

Joakim Sundnes Introduction to Scientific Programming with Python, Springer 2020

The learning outcomes (for the field of study and specialization)

K_W05: has extended knowledge in the field of the specialisation studied

K_W06: applies mathematics to the extent necessary to understand, describe and model chemical processes of extended complexity

K_U02: critically assesses the results of conducted, performed observations and theoretical calculations and discusses errors

K_U11: communicates in a foreign language in accordance with the requirements specified for level B2 of the Common European Framework of Reference for Languages and can use specialist terminology

K_K01: knows the limitations of her/his own knowledge; understands the need for further education and can inspire other people to do so

K_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

Knowledge

The student defines the concept of the algorithm. Names and describes the types and the data structures of the Python programming language. Differentiates the Python control instructions. Characterizes the most important classes of algorithms.

Skills

The student designs simple algorithms, writes them using the Python programming language and then tests the obtained programs.

Social competence

The student develops the skills of accurate and logical thinking and inference. 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. Develops the ability of working in a team.

Contact

cezary.czaplewski@ug.edu.pl