Python, Matlab for Geoscience

Course description sheet

Basic information

Field of study Geospatial Computer Science **Major** Remote Sensing and GIS **Organisational unit** Faculty of Geo-Data Science, Geodesy, and Environmental Engineering **Study level** Second-cycle (engineer) programme **Form of study** Full-time studies **Profile** General academic	**Didactic cycle** 2024/2025 **Course code** DGEITGS.IIi2.07205.24 **Lecture languages** English **Mandatoriness** Obligatory **Block** Major Modules **Course related to scientific research** Yes
Course coordinator	Mariusz Twardowski
Lecturer	Mariusz Twardowski

Period Semester 2	**Method of verification of the learning outcomes** Exam **Activities and hours** Lectures: 15 Project classes: 30	**Number of ECTS credits** 4

Goals

G1	The module allows recognizing possibilities of using Python as a tool for solving geoinformation problems.

Course's learning outcomes

Code	Outcomes in terms of	Learning outcomes prescribed to a field of study	Methods of verification
Knowledge – Student knows and understands:			
W1	rules of Python programming language	GEI2A_W02	Examination
Skills – Student can:			
U1	extend geoinformation tools functionality using Python language	GEI2A_U02	Examination
U2	create programs using Python language	GEI2A_U02	Examination
Social competences – Student is ready to:			
K1	creativity in programming	GEI2A_K03	Activity during classes
K2	language usage for application extension	GEI2A_K03	Activity during classes

Program content ensuring the achievement of the learning outcomes prescribed to the module

The module allows recognizing possibilities of using Python as a tool for solving geoinformation problems.

Student workload

Activity form	Average amount of hours* needed to complete each activity form
Lectures	15
Project classes	30
Preparation for classes	15
Realization of independently performed tasks	30
Examination or final test/colloquium	2
Contact hours	1
Preparation of project, presentation, essay, report	10
Student workload	**Hours** 103
Workload involving teacher	**Hours** 45

* hour means 45 minutes

Program content

No.	Program content	Course's learning outcomes	Activities
1.	Advanced scripts and program flow control. Script authoring tools refresh. Error correction and interpretation. Conditionals, loops and exception handling. Classes, objects, and modules. Class creation and method implementation. Inheritance examples. Object initialization and designator usage in method calls. Addidional module import. Common standard library examples. Spatial library usage. Raster, vector and text data operations. Data processing with selected libraries. Visualization libraries and Matlab API. Connecting PIL and Matplotlib libraries with QT interface. Usage of Matlab API in scripts. Usage of Numpy and SciPy libraries in geoscience problems. Parallel programming. Examples of paralel data processing. Coroutines and asynchronous I/O usage. Introduction to ML methods in geoscience. Some classes may be conducted as webinar.	U1, U2, K1, K2	Project classes
2.	Python basics rehash. Data types, their representation and complex structures. Advanced Scripts and program flow control. Conditional structures, loops and function definitions. Difference between interpreter and compiler. Exception handling. Input/output procedures. Classes, objects, and modules. Object-oriented programming in Python. Class, object and method construction, inheritance and polymorphism. Script modularisation methods and library import. Language libraries selection. Standard libraries and most important elements. Geospatial libraries GDAL, OGR, OSR. Reading, writing and processing raster and vector data. QT interface design for specific purposes. Rapid Application Development tools, automatic code generation, connecting interface elements with implementation code. Event-driven programming. QT GUI library. Matlab API for Python. Matlab Python modules. Extension trough PIL and matplotlib libraries. NumPy, SciPy libraries. Introduction to parallel programming concepts. Coroutines, asnchronous I/O and multithreaded process execution. Some classes may be conducted as webinar.	W1, K1	Lectures

Extended information/Additional elements

Teaching methods and techniques :

Remote learning trough UPEL and webinar tools., E-learning, Group work, Lectures

Activities	Methods of verification	Credit conditions
Lectures	Activity during classes, Examination	Positive grade
Project classes	Activity during classes, Examination	Positive grade

Conditions and the manner of completing each form of classes, including the rules of making retakes, as well as the conditions for admission to the exam

Classes are graded based on performing excercises and activity. Taking exam is conditioned by positive class grade. In case of a failed exam attempt, there will be possible 2 more tries. Retrying failed attempt does not have the impact on final grade. Preliminary requirement for the exam is to pass project classes.

Method of determining the final grade

To pass project classes assignment have to be fulfilled. Lectures and project classes final grade will be based on the class grade and final exam at the end of the semester.

Manner and mode of making up for the backlog caused by a student justified absence from classes

Student that is absent ought to learn material themselves.

Prerequisites and additional requirements

Ability to understand English.
Computer usage knowledge.
Basic Python language familiarity (f.e. DGI-2-107-TG-s course).
Ability to use a web browser.

Rules of participation in given classes, indicating whether student presence at the lecture is obligatory

Students participate in classes, covering the subject's content according to the syllabus. Students should continuously ask questions and clarify any doubts. Recording of the lecture requires the instructor's consent. Project classes: Students carry out practical work aimed at achieving the competencies outlined in the syllabus. The evaluation includes the execution of the project and the final outcome.

Literature

Obligatory

1. 1. Lutz Mark: “Learning Python, 5th edition”. 2013,
2. 2. Dawson Michael: “Python Programming for the Absolute Beginner, 3rd Edition”, 2010
3. 3. Documentation and online courses: http://pl.python.org
4. 4. Summerfield Mark: “Rapid GUI Programming with Python and Qt”. Prentice Hall 2008
5. 5. PyQt documentation: http://pyqt.sourceforge.net/Docs/PyQt5/

Scientific research and publications

Publications

1. Twardowski M., Pastucha E., Kolecki J., 2016: Performance of the automatic bundle adjustment in the virtualized environment
2. Hejamnowska B., Twardowski M., Żądło A., 2021: An application of the “traffic lights” idea to crop control in integrated administration control system
3. Rzonca A., Twardowski M., 2022, The lidargrammetric model deformation method for altimetric UAV-ALS data enhancement

Learning outcomes prescribed to a field of study

Code	Content
GEI2A_K03	aktywnego i kreatywnego włączenia się w dynamiczny rozwój geoinformacji, wzmacniania jej roli w społeczeństwie oraz popularyzowania powszechnego korzystania z danych przestrzennych.
GEI2A_U02	programować, modyfikować i rozbudowywać istniejące aplikacje oraz łączyć różne technologie informatyczne w zakresie geoinformacji.
GEI2A_W02	złożone zasady programowania oraz konstruowania algorytmów, niezbędne do czytania ze zrozumieniem, pisania, uruchamiania i weryfikacji programów.