Study program: Geoinformation Technologies

Qualification: PhD

General Information	
University	Yerevan State University (YSU)
Course title	**Geoprograming (Python)**
Course/Module code	Geoinformation Technologies
Course type	Mandatory
Year of Study	1st Year, 1st semester
Term/Semester	Autumn Semester
Credits awarded	2 (ECTS), 56 Hours (16 theory, 40 practice)
Degree	PhD
Enrollment status	Full-Time
Prerequisites and co-requisites (if applicable):	• General computer skills/Basic algorithm knowledge • Basic knowledge of database structure (Tables, keys, relationships), • Basic knowledge of Geoinformation technologies.

Lecturer's details	
Name, surname	Vahan Manukyan
Academic title	Associate Professor
Contact details	Email: v.manukyan@ysu.am,v.manukyanysu@yahoo.com
Office hours and consultation schedule	09:00-14:00 Monday, Wednesday, Friday

Course Structure	
Type (compulsory/ optional):	Compulsory
Course Goal	The aim of the course is to introduce the basic concepts of programming and scientific data analyzing by using Python programming language, develop coding skills and applying them to various GIS related problems.

	In this course, specific objectives are developing algorithmic solutions to geospatial problems, learning about solution strategies, high-level solution descriptions in pseudo-code, and translations of these into an implementation in Python programming language. By developing and reviewing code, students will be able to increase their knowledge in automation of geoprocessing.
Learning Outcomes	After completing this course, the students will be able: • To describe basic programming concepts, • To develop algorithms • To write codes in Python and visualize them in GIS (ArcPy), • To characterize data structure, and visualizing with “Matplotlib” library, • To develop functions with “Pandas” library and “Geopandas” module, • To describe reverse and forward addresses for geocoding,
Course contents	1. Introduction to Python in GIS, Python is a high-level, readable programming language known for its simplicity and versatility, suitable for various applications, including web development, data analysis, and automation. Many GIS software packages, such as ArcGIS and QGIS, support Python scripting. Users can write Python scripts to extend the functionality of these GIS tools, making them more customized to specific needs such as Data Manipulation, Automating Tasks, Map Creation and Spatial Analysis. 2. Basic concepts of Python and computer programming, In Python and computer programming basics, you start by grasping fundamental concepts like variables, which store data, and control structures like loops and conditionals. Functions, reusable blocks of code, are key, enabling modular and efficient program design. Understanding these basics forms the foundation for more advanced coding and problem-solving in the Python language.

3. Analyzing and visualizing Geodata with Python,

Analyzing and visualizing Geodata with Python involves using Python programming for the exploration and interpretation of geographic information. Python libraries like GeoPandas and Matplotlib enable the manipulation and representation of spatial data, allowing users to extract insights, perform geospatial analysis, and create visualizations such as maps and charts. This skill set is valuable for professionals working with location-based data in fields like geography, environmental science, and urban planning.

4. Variables and assignment,

In the context of Python programming, variables represent symbolic identifiers for data storage. Through the assignment operator '=', values are allocated to variables, establishing a link between the identifier and the assigned data. They provide the means to store, manage, and manipulate data within a program, forming the building blocks for more complex algorithms and logic. Variables allow programmers to work with different types of data, and assignments enable the updating and reassignment of values, essential for dynamic and responsive coding structures.

5. Operators and expressions,

Operators and expressions in Python are essential for performing computations and evaluations within a program. Operators are symbols that represent computations, such as addition or comparison, and they operate on operands, which can be variables or values. Expressions, composed of operators and operands, yield results when evaluated. Understanding how to use arithmetic, comparison, and logical operators, as well as constructing expressions, is fundamental for building effective and functional Python code.

6. Data Collections and data structures,

In Python, data collections and data structures are pivotal for organizing and managing information efficiently. Data collections, such as lists, tuples, sets, and dictionaries, allow the grouping of related data elements. These structures facilitate diverse operations, from sequential storage to key-value associations. Understanding and selecting appropriate data structures are critical for optimizing code performance and enabling streamlined data manipulation and retrieval in Python programming.

7. Introduction to ArcPy module

The ArcPy module serves as a crucial component in the realm of Geographic Information Systems (GIS), particularly within the Esri ArcGIS software.

Developed by Esri, ArcPy enables Python scripting for automating and extending GIS functionalities. It empowers users to manipulate spatial data, perform geoprocessing tasks, and automate workflows, fostering efficiency and customization in GIS applications.

8. Introduction of NumPy, Matplotlib, SciPy libraries,

NumPy, Matplotlib, and SciPy are fundamental Python libraries for scientific computing and data visualization. NumPy provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays efficiently. Matplotlib is a versatile plotting library used for creating static, animated, and interactive visualizations in Python. SciPy, built on NumPy, extends its capabilities by offering additional tools for optimization, signal processing, statistical analysis, and more. Together, these libraries form a powerful toolkit for scientific computing, data analysis, and visualization in Python.

9. Geometric Objects – Spatial Data Model,

In the context of spatial data modeling with Python, geometric objects play a central role in representing and manipulating spatial information. These objects, such as points, lines, and polygons, form the foundation of a spatial data model. Libraries like Shapely provide a rich set of tools for creating, analyzing, and performing geometric operations on these objects, facilitating the development of robust spatial applications and analyses in Python.

10. Geopandas (reading, writing a shapefile),

Geopandas is a Python library that simplifies working with geospatial data by extending the capabilities of Pandas, a popular data manipulation library. Geopandas facilitates the reading and writing of shapefiles, a common geospatial data format. By seamlessly integrating geometric operations with tabular data, Geopandas enables users to handle spatial information efficiently, making it a valuable tool for geospatial data analysis and manipulation in Python.

11. Geocoding in Geopandas,

Geocoding in Geopandas involves the process of converting textual location descriptions into geographic coordinates (latitude and longitude). Geopandas leverages geocoding services to assign spatial information to data, enabling the integration of location-based details into datasets. This functionality is valuable for mapping and spatial analysis, allowing users to work with diverse datasets by associating them with specific geographical locations.

	12. Retriving OpenStreetMap data The retrieval of OpenStreetMap data using Python entails the utilization of dedicated libraries, such as osmnx or overpy, to interact with the OpenStreetMap database. These tools facilitate the systematic extraction of geospatial information, encompassing features like street networks, points of interest, and structural footprints. This process empowers researchers and developers to integrate real-world, community-curated geographical data into Python workflows, fostering enhanced capabilities in mapping, geospatial analysis, and location-based applications.
Assessment methods and criteria	• Practical task • Project task • Final test exam For successfully accomplishment this course student must complete practical task with 5 points average grade. The project task with 10 points average grade which should be developed and introduced individually before final test exam. Project task should cover coding skills and working with geospatial data. Final exam will check the theoretical part of the course with 5 points average grade.
Recommended textbooks and links (in order of relevance):	• Python Documentation-https://docs.python.org/3/ • Zelle, J. (2010) Python Programming: An Introduction to Computer Science, Second edition. Franklin, Beedle & Associates. • Lawhead, J. (2015) Learning Geospatial Analysis with Python: An effective guide to geographic information systems and remote sensing analysis using Python 3, Second edition. Packt Publishing. • McKinney, W. (2012) Python for Data Analysis: Data wrangling with Pandas, NumPy and iPython, First edition. O´Reilly Media. • Westra, E. (2016) Python Geospatial Development: Develop sophisticated mapping applications from scratch using Python 3 tools for geospatial development, Third edition. Packt Publishing.

	• Zandbergen, P. (2013) Python Scripting for ArcGIS, Alternate edition. ESRI press. (Available from the library Diener, M. (2015) Python Geospatial Analysis Cookbook: Over 60 recipes to work with topology, overlays, indoor routing, and web application analysis with Python. • https://www.codecademy.com/ • http://www.coursera.org/ • http://www.datacamp.com/ • www.udemy.com • http://www.bostongis.com/ • http://www.esri.com/training • https://community.esri.com/groups/python