Syllabus
Course Program

Python Data Processing

Specialty 121 – Software Engineering 122 – Computer Science	Institute Institute of Computer Science and Information Technology
Educational program Software Engineering Computer Science and Intelligent Systems	Department Software Engineering and Management Intelligent Technologies (321)
Level of education Bachelor's level	Course type Special (professional), Elective
Semester 7	Language of instruction English, Ukrainian

Lecturers and course developers

First name and surname

Svitlana.Kovalenko@khpi.edu.ua
Candidate of Engineering Sciences, Associate Professor, Associate Professor of Software Engineering and Management Intelligent Technologies Department

Google Scholar:
https://scholar.google.com/citations?user=jeD1w74AAAAJ&hl
ORCID: https://orcid.org/0000-0001-6770-6778
Scopus: https://www.scopus.com/authid/detail.uri?authorId=57212035934
Web of Science: https://www.webofscience.com/wos/author/record/F-8252-2017)
More about the lecturer on the department's website

General information

Summary

The objective of the discipline is to acquire the knowledge, skills and abilities necessary for collecting, pre-processing, analyzing, visualizing, validating and cleaning data of various formats and types, working with APIs using built-in and third-party libraries of the Python programming language, as well as using data to make informed decisions

Course objectives and goals

Formation of students' theoretical and practical knowledge of effective work with data using the Python programming language, understanding of methods of data processing, analysis and visualization, as well as the use of data to make informed decisions in software development.

Format of classes

Lectures, laboratory classes, self-study, consultations. Final control in the form of a credit.

Competencies

121-К01. Ability to think abstractly, analyze and synthesize.
121-К02. Ability to apply knowledge in practical situations.
121-К05. Ability to learn and master modern knowledge.
121-К06. Ability to search, process and analyze information from various sources.
121-К13. Ability to identify, classify and formulate software requirements.
121-К15. Ability to develop architectures, modules and components of software systems.
121-К19. Knowledge of data information models, ability to create software for data storage, extraction and processing..
121-К22. The ability to accumulate, process and systematize professional knowledge in the creation and maintenance of software and recognize the importance of lifelong learning.
121-К23. Ability to implement phases and iterations of the life cycle of software systems and information technologies based on appropriate software development models and approaches.
121-К25. Ability to reasonably choose and master tools for software development and maintenance.
121-К26. Ability to think algorithmically and logically.
122- GC1. Ability to think abstractly, analyze and synthesise.
122- GC2. Ability to apply knowledge in practical situations.
122- GC6. Ability to learn and master modern knowledge.
122- GC7. Ability to search, process and analyze information from various sources.
122- GC8. Ability to generate new ideas (creativity).
122- GC11. Ability to make informed decisions.
122- PC8. Ability to design and develop software using various programming paradigms: generalised, object-oriented, functional, logical, with appropriate models, methods and algorithms of computation, data structures and control mechanisms.
122- PC9. Ability to implement a multi-level computing model based on client-server architecture, including databases, knowledge and data warehouses, to perform distributed processing of large data sets on clusters of standard servers to meet the computing needs of users, including cloud services.
122- PC10. Ability to apply methodologies, technologies and tools to manage the life cycle processes of information and software systems, information technology products and services in accordance with customer requirements.
122- PC12. Ability to ensure the organisation of computing processes in information systems for various purposes, taking into account the architecture, configuration, performance indicators of operating systems and system software.

Learning outcomes

121- PLO01. Analyze, purposefully search for and select information and reference resources and knowledge necessary for solving professional problems, taking into account modern achievements of science and technology.
121- PLO02. Know the code of professional ethics, understand the social significance and cultural aspects of software engineering and adhere to them in professional activities.
121- PLO08. Be able to develop a human-machine interface.
121- PLO12. Apply effective software design approaches in practice.
121- PLO13. Know and apply methods of developing algorithms, designing software and data and knowledge structures.
121- PLO15. Motivated to choose programming languages and development technologies to solve the problems of creating and maintaining software.
121- PLO18. Know and be able to apply information technologies for data processing, storage and transmission.
121- PLO23. Be able to document and present the results of software development.
122- PLO1. To apply knowledge of the basic forms and laws of abstract and logical thinking, the basics of the methodology of scientific knowledge, forms and methods of extracting, analyzing, processing and synthesizing information in the subject area of computer science.
122- PLO4. To use methods of computational intelligence, machine learning, neural network and fuzzy data processing, genetic and evolutionary programming to solve problems of recognition, forecasting, classification, identification of control objects, etc.

122- PLO10. To use tools for developing client-server applications, design conceptual, logical and physical models of databases, develop and optimize queries to them, create distributed databases, data warehouses and showcases, knowledge bases, including cloud services, using web programming languages.
122- PLO11. Have the skills to manage the life cycle of software, products and services of information technology in accordance with the requirements and restrictions of the customer, be able to develop project documentation (feasibility study, terms of reference, business plan, agreement, contract).
122- PLO20. Develop the architecture of software systems and their individual components in the construction of intelligent control systems in various industries, as well as manage the life cycle processes of software of intelligent control systems.

Student workload

The total volume of the course is 120 hours (4 ECTS credits): lectures - 16 hours, laboratory classes - 32 hours, self-study - 72 hours.

Course prerequisites

The study of this discipline is directly based on: "Object-oriented programming", "Fundamentals of programming", "Algorithmization and programming

Features of the course, teaching and learning methods, and technologies

Teaching and learning methods:
interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback, problem-based learning.

Forms of assessment:
written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS).

Program of the course

Topics of the lectures

Topic 1. Basic concepts of data processing. Basics of Python syntax. Development environments
The concept of data, the main tasks of data processing. Classification and general overview of data processing stages and methods. Installing Python. IDEs. Data input and output. Variables and data types: an overview of data types in Python, operations and functions performed on different types of data. Conditional statements and loops. Functions: defining and calling functions, parameters and arguments of functions, returning values from functions. Working with lists and strings. Handling of exceptions. Classes. Using built-in and third-party modules.
Topic 2. Jupytet notebook and NumPy library
Installation and basics of working with Jupytet notebook. Google Collaboratory. The history of NumPy creation. Arrays of ndarray. Comparison of NumPy arrays and Python lists. NumPy data types. Creating arrays. Operations on arrays. Universal functions. Concatenating and splitting NumPy arrays. Broadcasting. Comparisons, masks and Boolean logic. Fancy Indexing.
Topic 3. Data visualization
Libraries for data visualization: Matplotlib, Seaborn, Plotly. Matplotlib interfaces. Setting styles. Simple line graphs. Customizing graphs. Types of graphs. Visualization of errors. Creating multiple graphs in one figure.
Topic 4. Data processing with Pandas
Pandas objects. The Series and DataFrame properties. Building Pandas objects. Index object. Indexing and selecting data. Operations with data. Saving an index, Universal functions and index alignment. Operations between DataFrame and Series. Manipulation of data. Grouping of data. Processing of missing data. Hierarchical indexing. Combining datasets: concat(), append(), merge(), join(). Pivot tables. Visualization in Pandas.

Topic 5. Working with text

Processing text using the methods of the standard Python library. The re library and regular expressions. Templates and character classes. Quantifiers. Groups. String processing in Pandas. Vectorization of string operators. Pandas methods that use regular expressions

Topic 6. Working with graphics. PIL and opencv-python

Installing libraries. Representing images in libraries. Color models. Formats of graphic data. Basic manipulations with graphic data: resizing, cropping, transposing, rotating, inserting one image into another, creating watermarks. Detection of objects in the image. Pytesseract and text recognition.

Topic 7. Data collection. Requests and BeautifulSoup libraries

Library requests. GET and POST requests. Passing parameters, processing server response. JSON. Installing and importing BeautifulSoup. Loading a web page. Parsing an HTML page. Selecting elements and getting data. Navigating the page. Processing dynamic content. Selenium. Integration with Pandas. Saving data.

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Topic 1. Working in the Jupyter Notebook environment for data processing and visualization.

Topic 2. Working with data using Pandas tools.

Topic 3: Combining and processing data from different sources.

Topic 4. Processing text data with Pandas tools.

Topic 5. Processing of digital images.

Topic 6. Creating a web application for collecting and storing data using third-party APIs.

Self-study

Students are recommended additional materials (videos, articles) for self-study and processing.

Course materials and recommended reading

Key literature

1. Wes McKinney. Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter, 3rd edition, 2022, 540 p.
2. Jake VanderPlas. Python Data Science Handbook, 2d edition, O'Reilly Media, 2022, 550p.
3. Robert Johansson. Numerical Python. Scientific Computing and Data. Science Applications with Numpy, SciPy and Matplotlib. Second Edition, 2019, 685 p.
4. Ravishankar Chityala, Sridevi Pudipeddi. Image Processing and Acquisition using Python. CRC Press, 2021, 452 p.
5. Al Sweigart. Automate the Boring Stuff with Python, 2nd edition: Practical Programming for Total Beginners, No Starch Press, 2020, 901p.

Additional literature

1. Sandipan Dey. Hands-On Image Processing with Python: Expert techniques for advanced image analysis and effective interpretation of image data. Packt, 2018, 492 p.
2. Richard L. Halterman. Fundamentals of Python Programming, 2019, 669 p.
3. John Canning, Alan Broder, Robert Lafore. Data Structures & Algorithms in Python, Pearson Education, Inc., 2023, 1050 p.
4. Fabio Nelli. Python data analytics : with Pandas, NumPy, and Matplotlib, Apress, New York, NY, 2018, 555 p.
5. Takatomo Honda Flask Web Development from Scratch: Introduction to Developing Web Applications with Python, 2019, 137 p.

Assessment and grading

Criteria for assessment of student performance, and the final score structure

100% of the final grade consists of the results of the assessment in the form of a credit (40%) and current assessment (60%):
- 6 laboratory works (7% each);
- 2 tests (9% each).

Grading scale

Total points	National	ECTS
90–100	Excellent	A
82–89	Good	B
75–81	Good	C
64–74	Satisfactory	D
60–63	Satisfactory	E
35–59	Unsatisfactory (requires additional learning)	FX
1–34	Unsatisfactory (requires repetition of the course)	F

Norms of academic integrity and course policy

The student must adhere to the Code of Ethics of Academic Relations and Integrity of NTU "KhPI": to demonstrate discipline, good manners, kindness, honesty, and responsibility. Conflict situations should be openly discussed in academic groups with a lecturer, and if it is impossible to resolve the conflict, they should be brought to the attention of the Institute's management.
Regulatory and legal documents related to the implementation of the principles of academic integrity at NTU "KhPI" are available on the website: http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/

Approval

Approved by	08.06.2023	Head of the department Ihor HAMAIUN
	08.06.2023	Guarantors of the educational programs Andrii KOPP Uliya LITVINOVA