Syllabus
Course Program

Python Basic Programming Course

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
4

Language of instruction
English, Ukrainian

Lecturers and course developers

Svitlana Kovalenko

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 objectives of the discipline are to acquire knowledge, skills and abilities at the level of the latest achievements in the field of Python programming; study of basic concepts, mechanisms and techniques of procedural and object-oriented programming in Python; solving problems using various data structures, conditional structures, loops, functions, working with files, as well as skills in developing and debugging program code in compliance with best programming practices, acquiring skills in software design, development and testing

Course objectives and goals

Formation of students' theoretical and practical knowledge of the syntax of the Python programming language, the use of modules of its standard library, as well as the application of methods and techniques for creating, debugging and testing programs using various IDEs

Format of classes

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

Competencies

121-K01. Ability to think abstractly, analyze and synthesize.
121-K02. Ability to apply knowledge in practical situations.
121-K05. Ability to learn and master modern knowledge.
121-K06. Ability to search, process and analyze information from various sources.
121-K13. Ability to identify, classify and formulate software requirements.
121-K15. Ability to develop architectures, modules and components of software systems.
121-K19. Knowledge of data information models, ability to create software for data storage, extraction and processing..
121-K22. The ability to accumulate, process and systematize professional knowledge in the creation and maintenance of software and recognize the importance of lifelong learning.
121-K23. 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-K25. Ability to reasonably choose and master tools for software development and maintenance.
121-K26. Ability to think algorithmically and logically.
122- GC1. Ability to think abstractly, analyze and synthesis.
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 organization 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 150 hours (5 ECTS credits): lectures - 32 hours, laboratory classes - 32 hours, self-study - 86 hours.

Course prerequisites

The study of this discipline is directly based on: "Fundamentals of Web Development", "Databases", "Basic Python Programming Course", "Advanced Python Programming Course"

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: Introduction to Python. Basics of Python syntax, Zen of Python
Compilers and interpreters. Python as an interpreted programming language. Advantages and disadvantages of Python. Limits of application. History of creation. Versions of the language. How to install. IDES. Zen of Python. Key words. Case dependency and rules for building interpreters. PEP8. Expressions and operators. Sign of the end of the operator. Indents as a sign of a code block. The print() function. Python as a language with strong dynamic typing. Basic data types in Python. The concept of variable and immutable data types. Changes as pointers. Types of operators in Python
Topic 2. Comparison operators, logical operators, conditional statements
Comparators. Features of using comparison operators in Python. Logical operators and their features. Priority of operations. If statement, components of elif and else. Conditional expression. Walrus operator and its use. The match operator.
Topic 3: Loops in Python.
The while loop. Infinite loop and options for its use. The for loop and its features. The range() function. Break and continue statements. The else branch.
Topic 4. Lists in Python, operators, methods, and built-in functions for working with them. Investigating code performance
The concept of lists. Basic properties of Python lists. Basic operations with lists: adding, deleting, indexing items. Slices: concept, usage options. In and not in functions. List inclusions. The concept of function and method in Python. Built-in functions and methods for working with lists. Investigation of code performance.
Topic 5. User functions and working with them
Types of functions in Python. Syntax of the user function. Doc strings. Types of function parameters. The scope of the parameter. The global keyword. Recursion. Nested functions.

Topic 6. Strings in Python.
Convergence and difference between strings and lists. Ways to create string literals. Operators, methods and built-in functions for working with strings. String formatting language in Python. f-strings.
Topic 7. Tuples, dictionaries and sets. List, dictionary and set inclusions, generator expressions
Properties of tuples, their difference from lists, ways to create tuples. Operators, methods and built-in functions for working with tuples. The unpacking operation. The enumerate() operator. Dictionaries. Properties of dictionaries. Restrictions on dictionary keys. Dictionary elements and access to them. Methods of dictionaries. Properties of sets. Restrictions on the elements of the set. Operations with elements of sets. Mathematical operations with sets. Frozenset and its properties. Basics of working with list, dictionary and multiple inclusions and generator expressions.
Topic 8: Handling exceptions and the assert statement.
Exceptions and syntax errors. Built-in exceptions. Handling exceptions using try and except in Python. Branches of else and finally. User-defined exception handling. Assert statement and options for its use.
Topic 9: Lambda functions, map(), filter(), reduce() and zip() functions
The concept, creation and use of a lambda function. Examples of using map() and filter(). Using lambda functions in combination with map() and filter(). Options for using reduce() and zip().
Topic 10. Object-oriented programming in Python.
The concept of class and class instance. The keyword class. Method Resolution Order. Variables of a class instance. The self keyword. Class variables. Class, instance and static methods. Inheritance in Python. The concept of parent and child class. Types of inheritance. The super() function. Magic methods. @property decorator.

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Topic 1: Basics of working with the Jinja templating engine
Topic 2. Working with forms in Flask applications
Topic 3: Using third-party APIs in website development
Topic 4. Creating and deploying an online game using Flask
Topic 5. Creating web-oriented applications based on Django for solving linear programming problems
Topic 6 Working with Django models and deploying an application

Self-study

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

Course materials and recommended reading

Key literature

1. Luciano Ramalho. Fluent Python, O'Reilly Media, 2022, 980p
2. Eric Matthes Python Crash Course, 2nd Edition, No Starch Press, 2019, 548p.
3. Richard L. Halterman. Fundamentals of Python Programming, 2019, 669p.
4. Dusty Phillips, Steven F. Lott Python Object-Oriented Programming, 4th edition, Packt, 2019, 714p
5. J. Hunt: A Beginners Guide to Python 3 Programming. Springer, 2019, 433 p.

Additional literature

1. Al Sweigart. Automate the Boring Stuff with Python, 2nd edition: Practical Programming for Total Beginners, No Starch Press, 2020, 901p.
2. Al Sweigart. Python Programming Exercises, Gently Explained. 2022, 160p.
3. Allen B. Downey. Think Python 2e. https://greenteapress.com/wp/think-python-2e/
4. B. Stephenson: The Python Workbook: A Brief Introduction with Exercises and Solutions, 2 nd ed. // Springer, 2019
5. John Canning, Alan Broder, Robert Lafore. Data Structures & Algorithms in Python, Pearson Education, Inc., 2023, 1050p.

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%):
- 8 laboratory works (6% each);
- 2 tests (6% 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