

# PYTHON FRAMEWORKS

## COURSE SYLLABUS

<b>Code and name of specialty</b>	121 Software Engineering 122-Computer Science 126-Information Systems and technologies	<b>Institute / faculty</b>	Faculty of Computer Science and Software Engineering
<b>Program name</b>	Software Engineering Computer Science and Intelligent Systems Information Systems Software	<b>Department</b>	Software Engineering and Management Information Technologies
<b>Type of program</b>	Educational and Professional	<b>Language of instruction</b>	Ukrainian, English

### LECTURER

**Svitlana Kovalenko**

*Svitlana.Kovalenko@khti.edu.ua*



**Ph.D., Candidate of Engineering Sciences, Associate Professor, Associate Professor of Software Engineering and Information Technology Management. Has prepared and published more than 60 publications, 3 collective monographs, 2 textbooks, 6 articles in publications indexed in Scopus and Web of Science. (h-index = 3 in Google Scholar- <https://scholar.google.com/citations?user=jeD1w74AAAAJ&hl;> ID ORCID-0000-0001-6770-6778, Scopus author ID - [https://www.scopus.com /authid/detail.uri?authorId=57212035934](https://www.scopus.com/authid/detail.uri?authorId=57212035934)).**

**Leading lecturer of the courses:** *Python (Bachelors) (English and Ukrainian) Frameworks, Python (Bachelors) (English and Ukrainian) Data Processing, Fundamentals of Software Engineering (Bachelors) (in English)*

### GENERAL DESCRIPTION OF THE COURSE

<b>Summary</b>	The discipline "PYTHON FRAMEWORKS" is a discipline in the cycle of professional selective training in the specialty 121 "Software Engineering". It is taught in the fifth semester in the amount of 90 hours (3 ECTS credits), in particular: lectures - 16 hours, laboratory - 32 hours, independent work - 42 hours. The course provides two content modules and one module test. The discipline ends with a credit
<b>Course objectives</b>	Formation of students' theoretical and practical knowledge necessary developing Web-applications in the Python programming language using frameworks on the example of using a micro- and a full stack frameworks
<b>Types of classes and control</b>	Lectures, laboratory classes. Continuous assessment – laboratory works, intermediate modular assessment, course work. Final assessment – test.
<b>Term</b>	7

**Student workload (credits) / Type of course**

3 / elective

**Lectures (hours)**

16

**Workshops (hours)**

32

**Self-study (hours)**

42

**Program competences**

*General competencies:*

121-GC01. Ability to abstract thinking, analysis and synthesis.

121-GC02. Ability to apply knowledge in practical situations.

121-GC03. Ability to communicate in the state language both orally and in written form.

121-GC05. Ability to learn and master modern knowledge.

121-GC06. Ability to search, process and analyze information from various sources.

122-GC1. Ability to abstract thinking, analysis 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 justified decisions.

126-GC1. Ability to abstract thinking, analysis and synthesis.

126-GC2. Ability to apply knowledge in practical situations.

126-GC5. Ability to learn and master modern knowledge.

126-GC6. Ability to search, process and summarize information from various sources.

*Professional competencies of the specialty:*

121-PC13. Ability to identify, classify and formulate software requirements.

121-PC15. Ability to develop architectures, modules and components of software systems.

121-PC19. Knowledge of information data models, the ability to create software for data storage, retrieval and processing.

121-PC22. Ability to accumulate process and systematize professional knowledge on the creation and maintenance of software and recognition of the importance of lifelong learning.

121-PC23. Ability to implement phases and iterations of the life cycle of software systems and information technology based on appropriate models and approaches to software development.

121-PC25. Ability to reasonably select and master software development and maintenance tools.

121-PC26. Ability to algorithmic and logical thinking.

122-PC8. Ability to design and develop software using different programming paradigms: generalized, object-oriented, functional, logical, with appropriate models, methods and algorithms of calculations, data structures and management mechanisms.

122-PC9. Ability to implement a multi-tier computing model based on the client-server architecture, including databases, knowledge bases, and data warehouses, 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 according to customer requirements.

122-PC11. Ability to conduct intelligent data analysis based on methods of computational intelligence, including large and poorly structured data, their operational processing and visualization of analysis results in the process of solving applied problems.

122-PC12. Ability to ensure the organization of computational processes in information systems of various purposes, taking into account the architecture, configuration, performance indicators of operating systems and system software.

126-PC1. Ability to analyze the object of design or operation and its subject area.

126-PC3. Ability to design, develop, debug and improve system, communication and software and hardware of information systems and technologies, the Internet of Things (IoT), computer-integrated systems and system network structure, their management.

126-PC4. Ability to design, develop and use tools for the implementation of information systems, technologies and infocommunications (methodological, informational, algorithmic, technical, software and others).

126-PC10. Ability to select, design, deploy, integrate, manage, administer and maintain information systems, technologies and infocommunications, services and infrastructure of the organization.

126-PC12. Ability to manage and use modern information and communication systems and technologies (including those based on the use of the Internet).  
 126-PC13. Ability to perform computational experiments, compare the results of experimental data and solutions.

Learning outcomes	Teaching and learning methods	Forms of assessment (continuous assessment CAS, final assessment FAS)
121-PO01. Analyze, purposefully search for and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.	Interactive lectures with presentations, discussions, practical classes, teamwork, case method, research, project training	Written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), rapid surveys (CAS), online tests (CAS), final / semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
121-PO02. Know the code of professional ethics, understand the social significance and cultural aspects of software engineering and adhere to them in professional activities.		
121-PO12. Put effective approaches to software design into practice.		
121-PO13. Know and apply methods of algorithm development, software design and data and knowledge structures.		
121-PO15. Being motivated to choose programming languages and development technologies to solve problems of software design and maintenance.		
121-PO18. Know and be able to apply information technology processing, storage and transmission of data.		
121-PO23. Be able to document and present the results of software development.		
122-PLO1. Analyze, purposefully search for and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.		
122-PLO4. Use methods of computational intelligence, machine learning, neural network, and fuzzy data processing, genetic and evolutionary programming to solve problems of recognition, prediction, classification, identification of management objects, etc.		
122-PLO10. Use tools for developing client-server applications, design conceptual, logical, and physical models of databases, develop and optimize database queries, create distributed databases, repositories and showcases of databases, and knowledge bases, including those based on cloud services, using web programming languages.		
122-PLO11. Have the skills to manage the life cycle of software, products, and services of information technology under the requirements and restrictions of the customer, be able to develop project documentation (feasibility study, technical task, business plan, agreement, contract).		
122-PLO12. Apply methods and algorithms of computational intelligence and intelligent data analysis in the tasks of classification, prediction, cluster analysis, search for associative rules using software tools to support multidimensional data analysis based on technologies DataMining, TextMining, WebMining.		
122-PLO13. Know the system programming languages and methods for the software development that interacts with the components of computer systems, know network technologies, computer network architectures, have practical skills in administration technology of computer networks and their software.		

122-PLO20. Develop the architecture of software systems and their particular components during the construction of intelligent management systems in various fields, as well as manage the life cycle of intelligent management systems software.

126-PLO 2. Apply knowledge of basic and natural sciences, systems analysis and modeling technologies, standard algorithms and discrete analysis in solving problems of design and use of information systems and technologies.

126-PLO 3. To use basic knowledge of informatics and modern information systems and technologies, programming skills, technologies of safe work in computer networks, methods of creation of databases and Internet resources, technologies of development of algorithms and computer programs in high-level languages with application of project-oriented programming to solve problems of design and use of information systems and technologies.

126-PLO 5. Argue the choice of software and hardware for the creation of information systems and technologies based on the analysis of their properties, purpose and technical characteristics, taking into account the requirements for the system and operating conditions; have the skills to debug and test software and hardware of information systems and technologies.

126-PLO 6. Demonstrate knowledge of the current level of information systems technology, practical skills of programming and use of applied and specialized computer systems and environments for their implementation in professional activities.

126-PLO 7. Justify the choice of technical structure and develop appropriate software that is part of information systems and technologies.



**ASSESSMENT AND GRADING**

Range s of points corres pondi ng to grades	core (points) for all types of learning activities	ECTS grading scale	The national grading scale	Allocation of grade points
	90-100	A	excellent	
	82-89	B	good	
	74-81	C		
	64-73	D	satisfactory	
	60-63	E		
	35-59	FX		
	0-34	F	Unsatisfactory (with mandatory repetition of the course)	

**100% Final assessment** as a result of Final exam (30%) and Continuous assessment (70%).  
**30% Final test**  
**70% Continuous assessment:**  
 • 40% of assessment of tasks in laboratory works;  
 • 30% intermediate control (2 modular control works)

**Course policy** Students must attend all classes according to the study schedule and adhere to the norms of academic ethics. To study the course, students need to have their personal computer and (or) use computers of the computer center at the department. Students must work with compulsory and recommended reading, including Internet resources. Students must complete and submit all laboratory works during the semester in which the course is taught, before the examination session. The final assessment is not carried out without the personal presence of students.

**COURSE STRUCTURE AND CONTENT**

Topic 1	Basic concepts. Types of Python frameworks. Python web	Laboratory work 1	Create a WSGI application using Flask.	5 f- stu	Recover HTML, CSS and Javascript skills
---------	--	-------------------	--	----------	---

	frameworks and their main properties, WSGI, MVC paradigm			
Topic 2	Jinja template engine	Laboratory work 2	Basics of working with the Jinja template engine	Using the Bootstrap framework from Flask.
Topic 3	Web forms	Laboratory work 3	Create and submit forms using Flask	Detailed acquaintance with the module flask_wtf
Topic 4	Databases in Flask	Laboratory work 4	Website backend development based on RESTfull API	Detailed acquaintance with the request library. Processing the functionality of the application for testing the Postman API.
Topic 5	Introduction to the Django framework.	Laboratory work 5	Creation and adjustment of the Django project	Comparison of functionality and capabilities of known cloud platforms for hosting Web-applications created with Python frameworks
Topic 6	Getting started with Django	Laboratory work 6	Working with Django templates	Analysis of similarities and differences at the beginning of work with full stack and microframeworks
Topic 7	Models and work with them	Laboratory work 7	Working with Django models	Types of relationships between models, model validation and validators. Comparison of functionality and limits of application of micro- and full stack frameworks
Topic 8	Branching of access and REST Web-services	Laboratory work 8	Work with Web application users and create REST API	Analysis of Python frameworks for testing Web applications

### RECOMMENDED READING

<b>Compulsory</b>	1. Гринберг, М. (2014). Разработка веб-приложений с использованием Flask на языке Python. ДМК Пресс, 272 с.	<b>Recommended</b>	5. Grinberg, M. (2018). Flask Web Development. O'Reilly Media Inc., 314 p.
	2. Дронов, В. А. (2019). Django 2.1: практика создания Web-сайтов на Python. Санкт-Петербург: БХВ-Петербург, 672 с.		6. Jaiswal, S., Kumar, R. (2014). Learning Django Web Development. Packt Publishing, 336 p.
	3. Чан, У. (2015). Python: создание приложений. Библиотека профессионала, 3-е изд.: пер. с англ. Москва: ООО «И. Д. Вильямс», 816 с.		7. Marani, F. (2019). Practical Django 2 and Channels 2 Building Projects and Applications with Real-Time Capabilities. Apress, 419 p.
	4. Rubio, D. (2017). Beginning Django. Web Application Development and Deployment with Python. Apress, 609 p.		8. Flask мера: textbook. [Electronic resource]. Access mode: <a href="https://habr.com/ru/post/346306/">https://habr.com/ru/post/346306/</a>
			9. Посібник по Django для початківців. [Електронний ресурс]. Режим доступу: <a href="https://codeguida.com/post/1039">https://codeguida.com/post/1039</a>
			10. Flask Tutorial. [Electronic resource]. Access mode: <a href="https://flask.palletsprojects.com/en/2.0.x/tutorial/">https://flask.palletsprojects.com/en/2.0.x/tutorial/</a>

### Academic integrity

Graduate students are expected to adhere to the Code of Ethics of Academic Relations and Integrity” of NTU “KhPI”.

The content of this syllabus is consistent with the course program.