National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

Computer Systems
Software Department

# PYTHON APPLIED SOFTWARE DEVELOPMENT

## Syllabus

### Requisites of the Course

| | |
|---|---|
| **Cycle of Higher Education** | ***First cycle of higher education (Bachelor's degree)*** |
| **Field of Study** | *12 Information Technologies* |
| **Speciality** | *121 Software engineering* |
| **Education Program** | *Software Engineering of Multimedia and Information Retrieval Systems* |
| **Type of Course** | *Elective* |
| **Mode of Studies** | *full-time* |
| **Year of studies,semester** | *3 year ( 6 semester)* |
| **ECTS workload** | *36 hours for lectures, 18 hours for practicals, 66 hours for self-study* |
| **Testing and assessment** | *Credit test* |
| **Course Schedule** | *Classes by the timetable http://rozklad.kpi.ua/* |
| **Language of Instruction** | *English* |
| **Course Instructors** | *Lecturer: PhD, Associate Professor, Dmytro Novak, novak.knutd@gmail.com*<br>*Teacher of computer workshop: PhD, Associate Professor, Dmytro Novak, novak.knutd@gmail.com* |
| **Access to the course** | *Google classroom: Access is given to registered students.* |

### Outline of the Course

#### 1. Course description, goals, objectives, and learning outcomes

*The study of the discipline "Python Applied Software Development" allows students to develop the competencies necessary for solving practical problems of professional and scientific activities related to the preparation and development of software in the Python language.*

*The **purpose** of studying the discipline "Python Applied Software Development" is to provide students with knowledge and practical skills in the field of software development using the Python language.*

*The **subject** of the discipline "Python Applied Software Development" is study the basics of programming in the Python language and use it to create various software products and solve practical problems.*

*The study of the discipline "Python Applied Software Development" contributes to the formation of the following **professional competence (PC)** in students according to the educational program:*

***PC8** Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.*

*The study of the discipline "Fundamentals of computer systems and networks" contributes to the formation of the following **program learning outcomes (PLO)** for students according to the educational program:*

***PLO01** To analyze, purposefully search and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.*

***PLO07** To know and to apply in practice the fundamental concepts, paradigms and basic principles of the functioning of language, instrumental and computational tools of software engineering.*

***PLO18*** *To know and be able to apply information technology of processing, storage and transmission of data.*

## 2. Prerequisites and post-requisites of the course (the place of the course in the scheme of studies in accordance with curriculum)

*Successful study of the discipline "Python Applied Software Development" is preceded by the study of disciplines: "Programming basics", "Programming" and "Databases" of the curriculum for bachelors in the specialty 121 Software Engineering.*

*The theoretical knowledge and practical skills obtained during the mastering of the discipline "Python Applied Software Development" contribute to the successful completion of course projects and bachelor's thesis.*

*To successfully master the discipline requires a basic level of English not less than A2.*

## 3. Content of the course

*Discipline "Python Applied Software Development" involves the study of the following topics:*

*Topic 1. Fundamentals of Python programming.*

*Computer Workshops 1-3.*

*Topic 2. Peculiarities of software development in Python.*

*Credit test.*

## 4. Coursebooks and teaching resources

***Basic references:***


1. *Sutor R. S. Dancing with python: Learn python software development from scratch and get started with quantum computing. – Packt Pub. 2021, 744 p.*
2. *Slatkin B. Effective Python. – Addison-Wesley Professional. 2019, 480 p.*


***Additional references**:*


3. *Flask. Electronic resource. Access mode: https://www.tutorialspoint.com/flask/index.htm*
4. *Pillai A. B. Software architecture with Python. – Packt Publishing Ltd. 2017, 619 p.*
5. *Bird A., Han L. C., Jiménez M. C., Lee G., Wade C. The The Python Workshop: Learn to code in Python and kickstart your career in software development or data science. – Van Haren Publishing. 2019, 608 p.*
6. *Wilkes M. Advanced Python Development: Using Powerful Language Features in Real-World Applications (1st ed.). – Apress. 2020, 628 p*
7. *Shaw B., Badhwar S., Bird A. Web Development with Django. Van Haren Publishing. 2021, 826 p.*


**Educational content**

## 5. Methodology

| № | Type of training session | Description of the lesson |
|---|---|---|
| | *Topic 1. Fundamentals of Python programming.* | |
| 1 | *Lecture 1. Python basics.* | *History and features of the Python language. Basic concepts of the Python language.* |
| 2 | *Lecture 2. Data types and structures.* | *Logical operators. Cycles. Lists. Tuples Rows Dictionaries.* |
| 3 | *Lecture 3 Exception handling and data logging.* | *Exceptions Exception generation in Python. Own exceptions. Logging in. Data logging. Exception logging.* |

| | | |
|---|---|---|
| 4 | Lecture 4. Working with functions. | Functions. Arguments of functions. Function tests. Functions as objects. Function scope. Local and global variables. Anonymous functions. Locking. Decorators. |
| 5 | Computer Workshop 1. Working with lists, dictionaries, loops and built-in functions. | Objective: Demonstrate your skills in working with lists and dictionaries, learn basic methods for working with them; learn types of loops in Python; consider the main built-in functions of the Python programming language and learn how to work with them. |
| 6 | Lecture 5. Modules and packages. | Python code in normal and interactive mode. CPython interpreter. Modules in Python. Using packages in Python. |
| 7 | Computer Workshop 2. Working with iterators and generators. | Objective: learn the concepts of iterator and generator in Python, as well as their advantages; familiarize yourself with their use. |
| 8 | Lecture 6. Work with files. | Writing data to a file. Creating and deleting directories. Statement with. Structured text files. Characteristics of XML and JSON formats. |
| 9 | Computer Workshop 3. Working with files and formatting data in JSON format. | Objective learn how to work with files using functions from the standard library; consider the concept of text parsing and familiarize yourself with its use in Python; explore the possibilities of interaction of Python with the JSON data storage format. |
| Topic 2. Peculiarities of software development in Python. | | |
| 10 | Lecture 7. An introduction to object-oriented programming in Python. | Abstraction, encapsulation, inheritance and polymorphism. Creating a class and its attributes. |
| 11 | Lecture 8. Fundamentals of object-oriented programming in Python. | Object methods and class methods. Encapsulation (Public, Protected, Private). Inheritance and polymorphism. Multiple inheritance. Class hierarchy design and object class. |
| 12 | Lecture 9. Deep learning of object-oriented programming in Python. | Mixins in Python. Aggregation. Iterators and generators. Coroutines Singleton. Metaclasses. Dynamic creation of classes. |
| 13 | Lecture 10. Unit testing in Python. | Frameworks for autonomous testing in Python. The main structural elements of unit-test. Command line interface. Downloading and running tests. Testing by categories. Combination of tests. |
| 14 | Lecture 11. Introduction to Flask. | Contexts in Flask. Request processing. Server response. Interception of requests. Templates. |
| 15 | Lecture 12. Flask basics. | Basics of the Jinja Template Builder. Comments. Declaration of variables. Filters. Macros. Screening. Creating URLs in Flask. Working with static files in Flask. Automatic import of objects. |
| 16 | Lecture 13. Deep learning of Flask. | Working with forms in Flask. Forms in the console. Form visualization. Work with form confirmation. Configuring cookies in Flask. Creation of models. Deploying a Flask application. |
| 17 | Lecture 14. The Numpy library | Arrays. Operations on arrays. Array modification. Examples of working with Numpy. |
| 19 | Lecture 15. Matplotlib library | Basic graphics commands. Text parameters of graphs. Graphics window architecture. Placement of |

| | | |
|---|---|---|
| | | *graphs. Ways to display the legend. Saving graphs to a file.* |
| *20* | *Lecture 18. Final lecture.* | *Credit Test.* |

### 6. Self-study

*The discipline "Python Applied Software Development" is based on independent preparation for classroom classes on theoretical and practical topics.*

| *№* | *The name of the topic that is submitted for independent study* | *Hours of study* | *References* |
|---|---|---|---|
| *1* | *Preparing for Lectures* | *27* | *[1-7]* |
| *2* | *Preparing for Computer Workshops 1-3.* | *21* | *[1, 2]* |
| *3* | *Preparing for Test 1 (Topic 1).* | *4* | *[1-4]* |
| *4* | *Preparing for Test 2 (Topic 2).* | *4* | *[2-7]* |
| *5* | *Preparing for Credit Test.* | *10* | *[1-7]* |

## Policy and Assessment

### 7. Course policy

- *Attendance at lectures is mandatory.*
- *Attendance at computer workshops can be sporadic and if you need to defend a computer workshop.*
- *Rules of conduct in the classroom: activity, respect for those present, turning off the phones.*
- *Adherence to the policy of academic integrity.*
- *Rules for protecting the work of the computer workshop: the work should be done according to the option of the student, which is determined by his number in the group list.*
- *The rules for assigning penalty points are as follows.*

### 8. Monitoring and grading policy

*During the semester, students complete 3 computer workshops. Maximum number of points for each computer workshop: 20 points.*

*Points are awarded for:*

*- quality of computer workshop: 0-8 points;*

*- answer during the defense of the computer workshop: 0-8 points;*

*- timely submission of work to the defense: 0-4 points.*

*Maximum number of points for performing and defending computer workshops:*

*20 points × 3 comp. work. = 60 points.*

*The task for the modular test consists of 5 questions - 3 theoretical and 2 practical. The answer to each theoretical question is evaluated by 8 points, the answer to the practical question is evaluated by 8 points.*

*Criteria for evaluating each theoretical question of the module test:*

*7-8 points - the answer is correct, complete, well-argued;*

*4-7 points - there are minor errors in the answer;*

*1-3 points - there are significant errors in the answer;*

*0 points - no answer or the answer is incorrect.*

*Criteria for evaluating the practical question of modular control work:*
*7-8 points - the answer is correct, complete, well-argued;*
*4-7 points - there are minor errors in the answer;*
*1-3 points - there are significant errors in the answer;*
*0 points - no answer or the answer is incorrect.*
*Maximum number of points for each modular test:*
*8 points × 3 theoretical questions + 8 points × 2 practical questions = 40 points.*

*The rating scale for the discipline is equal to:*
*R = Rs = 60 points + 40 points = 100 points.*
*According to the description:* $R = R_{comp.work} + R_{Credit\ test} = 60 + 40$ *points = 100 points*

*Calendar control: conducted twice a semester as a monitoring of the current state of compliance with the requirements of the syllabus.*
*At the first attestation (8th week) the student receives "credited" if his current rating is not less than 50% of the maximum number of points that the student can receive before the first attestation.*
*At the second attestation (14th week) the student receives "credited" if his current rating is not less than 50% of the maximum number of points that the student can receive before the second attestation.*

*Semester control: Credit Test.*

*Conditions of admission to semester control:*
*With a semester rating (Rs) of at least 60 points and enrollment in all computer workshops, the student receives the final test"automatically" according to the table (Table of correspondence of rating points to grades on the university scale).*
*Prerequisite for admission to the final test is the implementation and defense of a computer workshop.*
*The final performance score or the results of Final test the Fail/ Pass are adopted by university grading system as follows:*

| *Score* | *Grade* |
|---|---|
| *100-95* | *Excellent* |
| *94-85* | *Very good* |
| *84-75* | *Good* |
| *74-65* | *Satisfactory* |
| *64-60* | *Sufficient* |
| *Below 60* | *Fail* |
| *Course requirements are not met* | *Not Graded* |

**Syllabus of the course**

**Is designed by teacher** PhD, Associate Professor, Dmytro Novak

**Adopted by** Computer Systems Software Department (protocol № 8 from 25.01.23)

**Approved by** the Faculty Board of Methodology (protocol № 6 from 27.01.23)