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

Computer Systems Software Department

NETWORK SOFTWARE DESIGN AND DEVELOPMENT

Syllabus

Requisites of the Course	
Cycle of Higher Education	***Second cycle of higher education (Master's degree)***
Field of Study	*12 Information Technologies*
Speciality	*121 Software engineering*
Education Program	*Software engineering of multimedia and information-search systems*
Type of Course	*Selective*
Mode of Studies	*full-time*
Year of studies, semester	*2 year (3 semester)*
ECTS workload	*4 credits (ECTS). Time allotment - 120 hours, including 46 hours of classroom work (Lectures: 32 hours, Laboratory work: 14 hours), and 74 hours of self-study.*
Testing and assessment	*Final test, modular test , calendar control*
Course Schedule	*Classes by the timetable http://roz.kpi.ua/*
Language of Instruction	*English*
Course Instructors	*Lecturer: PhD, Associate Professor, Liubov Oleshchenko, oleshchenkoliubov@gmail.com* *Laboratory work: PhD, Associate Professor, Liubov Oleshchenko, oleshchenkoliubov@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 "Network Software Design and Development" allows students to form the competencies necessary to solve practical problems of professional activities related to the design and development of network software. The **purpose** of studying the discipline "Network Software Design and Development" is the formation of students' ability to develop software for centralized administration of a computer network in accordance with certain requirements; programmatically configure IoT devices for their operation in a network of a given topology. The **subject** of the discipline "Network Software Design and Development" are technologies for programming IoT devices and software methods of SDN network administration.*

After mastering the discipline "Network Software Design and Development" learning outcomes are:

knowledge:

- protocols and IoT standards;

- IoT network architectures;

- software methods of data protection of IoT networks;

- network programmability concepts using SDN, OpenFlow Controller, NFV, NETCONF, RESTCONF, Orchestration, YANG, YAML, ACI, APIC-EM and containers technologies.

skills:

- design networks and program IoT devices in the Packet Tracer modeling environment and perform testing of programmed IoT devices in a network of a given topology.

experience:

- design of IoT networks;

- development of software for IoT devices for their operation in a network of a given topology.

The study of the discipline "Network Software Design and Development" contributes to the formation of students of higher education who study under the educational program "Software Engineering of Multimedia and Information Retrieval Systems" competencies necessary for solving practical problems of professional activity related to the use programming technologies for building IoT and SDN networks:

GC01 *Ability to abstract thinking, analysis and synthesis.*

GC03 *Ability to conduct research at the appropriate level.*

PC02 *Ability to develop and implement scientific and / or applied projects in the field of software engineering.*

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 "Network Software Design and Development" is preceded by the study of the disciplines "Operating Systems", "Programming" and "Computer Systems and Networks Fundamentals" curriculum for bachelors in 121 "Software Engineering".

The theoretical knowledge and practical skills obtained during the mastering of the discipline "Network Software Design and Development" can be useful for further study in graduate school in the specialty 121 "Software Engineering".

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

3. Content of the course

Discipline "Network Software Design and Development" involves the study of the following topics:

Topic 1. Design and development of software for IoT devices.

Topic 2. Software development for SDN networks.

Modular test.

Final test.

4. Coursebooks and teaching resources

Basic references:

1. Finardi A. IoT Simulations with Cisco Packet Tracer. https://www.theseus.fi/bitstream/handle/10024/150158/Andrea%20Finardi%20%20Master%20of%20Engineering%20%20Information%20technology.pdf?sequence=1&isAllowed=y.

2. Leading the IoT. https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf.

3. Changing the programming paradigm for the embedded in the IoT domain. http://ieeexplore.ieee.org/document/7389059/?arnumber=7389059.

4. Things and Components available in Packet Tracer 7.2. https://www.packettracernetwork.com/internet-of-things/pt7-iot-devices-configuration.html.

***Additional references**:*

1. The basics of SDN and the OpenFlow Network Architecture. https://noviflow.com/the-basics-of-sdn-and-the-openflow-network-architecture.

2. What is SDN and where software-defined networking is going. https://www.networkworld.com/article/3209131/what-sdn-is-and-where-its-going.html.

Educational content

5. Methodology

No.	Type of training session	Description of the training session	Hours
Topic 1. Design and development of software for IoT devices.			
1	*Lecture 1. Internet of Things. IoT protocols and standards.*	*Internet of Things. IoT protocols and standards. IoT World Forum Reference Model.*	*2*
2	*Lecture 2. IoT support in industry. Industrial IoT devices. Data protection and control layers in IoT.*	*IoT support in industry. Industrial IoT devices. Data protection and control layers in IoT.*	*2*
3	*Lecture 3. Connecting IoT devices to the network.*	*Connecting IoT devices to the network. Sensors, controllers. Actuators. IP controllers. IoT software examples.*	*2*
4	*Laboratory work 1. Connecting IoT devices to smart home.*	*Task: in the Packet Tracer simulation environment, create a smart home network according to the instructions, add Wired IoT wireless devices to the Smart Home Network, configure the network adapter, and check the status of the devices via IoT Server.*	*2*
5	*Lecture 4. Hardware and software components of the Arduino computing system.*	*Hardware and software components of the Arduino computing system. Arduino software structure.*	*2*
6	*Lecture 5. Cloud and fog computing for IoT devices.*	*Cloud and fog computing for IoT devices. Software development for monitoring crop status on agricultural plantations.*	*2*
7	*Lecture 6. IoT architectural models.*	*IoT architectural models: Device-to-Device, Device-to-Cloud, Device-to-Gateway-to-Cloud, Device-to-Gateway-to-Cloud-to-Application.*	*2*
8	*Laboratory work 2. Monitoring IOT devices on the network.*	*Lesson 1. Connecting a home gateway and IoT devices. Task: In a Cisco Packet Tracer environment, connect the Home Gateway to the network. Configure the wireless network (SSID, password) and connect several IoT devices (e.g., lamps, sensors, camera) to it. Verify that the devices are successfully connected and have IP addresses.*	*2*
		Lesson 2. Connecting the end user device and monitoring IoT. Task: Add a user's end device (laptop or	*2*

		smartphone) to the network and connect it to the Home Gateway via Wi-Fi. Monitor IoT devices using the Home Gateway interface: check their status, change settings (on/off, change parameters), and track the connection between them.	
9	*Lecture 7. Raspberry Pi architecture. Prototyping Lab Application.*	*Raspberry Pi architecture. Raspberry Pi bootable SD card. Prototyping Lab Application.*	*2*
10	*Laboratory work 3. Fog computing in a smart home.*	*Task: to investigate a smart home, to use fog computing in a smart home for a computing system to monitor the impact of smoke levels detected in the home.*	*2*
11	*Lecture 8. Anatomy of IoT attacks. Types of IoT attacks. Demonstration of the use of Mirai Botnet.*	*Anatomy of IoT attacks. Types of IoT attacks. Demonstration of the use of Mirai Botnet.*	*2*
12	*Lecture 9. IoT Security Model. NICE Cybersecurity Workforce Framework.*	*IoT Security Model. NICE Cybersecurity Workforce Framework. MQTT Security.*	*2*
13	*Laboratory work 4. Configuring security for IoT devices.*	*Task: In the Cisco Packet Tracer environment, build a network with a wireless router and several IoT devices (smart lamps, sensors, cameras). Configure the SSID, access password, and Wi-Fi encryption type (WPA2-PSK). Make sure that IoT devices connect successfully only after entering the correct access key. In the same topology, activate access control mechanisms (MAC filtering, user settings, rights restrictions). Perform a security check: try to connect an unauthorized device and make sure that it does not have access to the network. Monitor connected IoT devices through the router interface and verify that only authorized devices are functioning.*	*2*

14	*Laboratory work 5. Designing a network for a smart home and programming IoT devices.*	*Task: in Cisco Packet Tracer, build a "smart home" network topology according to a given scenario (Home Gateway, WLAN, IoT devices, user PC). Configure a DHCP server to automatically issue IP addresses, activate WLAN for wireless connection of IoT devices, add a DNS server that translates the URL of the IoT home page to the corresponding IP address of the IoT server. Program IoT devices (smart lamps, motion sensors, cameras, smart doors, etc.) for their interaction with each other and with the network. Test the operation of the devices: check the availability of the IoT home page via a PC browser, the correct operation of DHCP and DNS, as well as the interaction of IoT devices in the "smart home" scenario.*	*2*
Topic 2. Software development for SDN networks.			
15	*Lecture 10. SDN network architecture.*	*History of SDN networks. Basic concepts and architecture of SDN networks.*	*2*
16	*Lecture 11. OpenFlow Technologies. OpenFlow Software*	*Purpose of OpenFlow in SDN networks. Table of OpenFlow actions and states. OpenFlow software. Switch Software. Controller Plane Software. Examples of using OpenFlow software Switch Software, Controller Plane Software.*	*2*
17	*Lecture 12. Software for topology monitoring, traffic analysis, latency, and routing in an SDN network .*	*Software for topology monitoring, traffic analysis, latency and routing in SDN networks. Software testing in Mininet Network Emulator.*	*2*
18	*Lecture 13. Software development for displaying the host table and network device table in an SDN network.*	*Develop software to query and display a table of hosts and a table of network devices on a network. Inventory network hosts in Python. Create a function to query a host inventory. Analyze the Network Device Inventory API using Python.*	*2*
19	*Lecture 1 4. Purpose of APIC-EM. Using Postman to interact with REST API.*	*Purpose of APIC-EM. Using Postman to interact with REST API.*	*2*
20	*Lecture 15. Development of network software for interaction with APIs.* *JSON Parsing in Python.* *MapQuest API Application.* *RESTful Request Authentication.*	*Python programs for user input, reading and writing to external files. Python programs that access APIs based on user input to display data in JSON format. Parsing JSON in Python. MapQuest API Application. RESTful Request Authentication.*	*2*
2 1	*Laboratory work 6. Software development to retrieve data from the MapQuest Directions API.*	*Task: develop software to retrieve data in JSON format from the MapQuest Directions API using the Python language.*	*2*
24	*Lecture 16. Final lesson.*	*Repetition of studied material. Modular test work.*	*2*

6. Self-study

The discipline "Network Software Design and 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 Laboratory works 1-6.	12	[1-4]
2	Preparing for Topic 1.	18	[1-2]
3	Preparing for Topic 2.	12	[3-4]
4	Preparing for Final test.	32	[1-4], add. [1-2]

Policy and Assessment

7. Course policy

Forms of organizing the educational process, types of training sessions and assessment of learning outcomes are regulated by the Regulations on the Organization of the Educational Process at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

***Class attendance policy**. Attendance at lecture classes is mandatory. Attendance at laboratory classes may be occasional and required for the defense of laboratory work. The presence or absence of a student at a class is not assessed by awarding or deducting points. If a student cannot attend classes, he or she is still responsible for studying the theoretical material and completing practical assignments.*

***Policy on ethical norms in the classroom**: discipline; compliance with subordination; honesty; responsibility; respect for those present, turning off phones.*

***Policy on assessing learning outcomes**. The policy on assessing learning outcomes is regulated by the Regulations on the system of assessing learning outcomes at Igor Sikorsky Kyiv Polytechnic Institute. According to the Regulations, each grade is given in accordance with the criteria developed by the teacher and announced to students in advance. If a student fails to complete all four laboratory tests, he/she will not be allowed to take the test. Failure to pass the current control measure (modular test) without good reason is assessed as 0 points.*

The policy and principles of academic integrity are regulated by the norms set of the Code of Honor of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" (https://kpi.ua/files/honorcode_en.pdf); Regulations on the Organization of the Educational Process, Regulations on the System for Preventing Academic Plagiarism, Regulations on the Commission on Ethics and Academic Integrity. Plagiarism and other forms of violation of the principles of academic integrity are unacceptable. The student must complete all laboratory practical tasks independently using open sources of information and acquired knowledge and skills.

Plagiarism and other forms of violation of the principles of academic integrity are unacceptable. All work for current and semester tests must be completed independently by the student using open sources of information and the acquired knowledge and skills.

All works that violate the principles of academic integrity (the program code does not match the assignment option, the identity of the program code among different works, etc.) are evaluated at 0 points. To gain access to the test, the student must independently complete the laboratory work (without changing the current rating). In the case of semester control work that violates the principles of academic integrity, the semester control report is marked "Eliminate".

Policy on appealing the results of assessment of control measures. According to the "Regulations on resolving conflict situations at Igor Sikorsky Kyiv Polytechnic Institute" (https://osvita.kpi.ua/sites/default/files/downloads/regulations_resolving_conflict_situations_2020.pdf) , students have the right to appeal the results of control measures with arguments, explaining which

criterion they disagree with according to the assessment. A student can raise any issue related to the procedure of control measures and expect that it will be considered in accordance with predetermined procedures.

Policy on assigning incentive points. According to the "Regulations on the system of assessing learning outcomes at Igor Sikorsky Kyiv Polytechnic Institute", incentive points are not included in the main RSO scale, and their sum cannot exceed 10 points.

Incentive points are awarded for a creative approach in performing laboratory work (the maximum number of points for all work is 10 points), as well as for participation in scientific projects and conferences related to the topic of this course.

8. Monitoring and grading policy

During the semester, students perform 6 laboratory works. The maximum number of points for each laboratory work: 10 points.

Points are awarded for:

- quality of laboratory work: 0-4 points;

- answer during the defense of laboratory work: 0-4 points;

- timely presentation of work for defense: 0-2 points.

Performance evaluation criteria:

4 points – the work is done qualitatively, in full;

3 points – the work is done qualitatively, in full, but has shortcomings;

2 points – the work is done qualitatively, but not in full, has flaws;

1 point – the work is incomplete or contains errors.

0 points – the work contains significant errors.

Answer evaluation criteria:

2 points – the answer is complete, well-argued;

1 point – there are significant errors in the answer;

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

Criteria for evaluating the timeliness of work submission for defense:

2 points – the work is presented for defense no later than the specified deadline;

0 points – the work is submitted for defense later than the specified deadline.

The maximum number of points for the performance and defense of laboratory work:

10 points × 6 lab. = 60 points.

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

Evaluation criteria for each theoretical/practical question of the modular test:

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

5-6 points – the answer is correct, but incomplete or poorly argued;

3-4 points – there are minor errors in the answer;

1-2 points – there are significant errors in the answer;

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

The maximum number of points for a modular control work:

8 points × 3 theoretical questions + 8 points × 2 practical questions = 40 points.

The rating scale for the discipline is equal to:

R = R s = 60 points + 40 points = 100 points.

According to the description: $R = R_{lab} + R_{mod.test}$ = 60+40 points = 100 points

Calendar control: is carried out twice a semester as a monitoring of the current state of fulfillment of the syllabus requirements.

At the first attestation (8th week), the student receives "credited" if his current rating is at least 50% of the maximum number of points that the student can receive before the first certification (20 points). At the second attestation (14th week), the student receives "passed" if his current rating is at least 50% of the maximum number of points that the student can receive before the second certification (30 points).

Semester control: ***Final test****.*

Conditions for admission to semester control: with a semester rating (R_S) of not less than 60% (60 points) and the enrollment of all laboratory works.

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*

9. Additional information about the course

According to the Regulations on the recognition of learning outcomes acquired in non-formal/informal education at Igor Sikorsky Kyiv Polytechnic Institute https://osvita.kpi.ua/node/179, during the study of the discipline, it is possible to enroll in an online course or other element of non-formal education in the discipline.

A certificate of completion of a similar course is valued at 10 points (if the course topic matches the lab topic), writing articles or participating in conferences/projects on the relevant topic of the discipline is also valued at an additional 5 points. For example, a lab can be automatically credited as 10 points by sending a certificate of completion of a Coursera course on the topic of the relevant lab to the classroom, for example:

An Introduction to Programming the Internet of Things (IoT) Specialization — UC Irvine

A six-course series teaching the design, creation, and deployment of IoT devices using Arduino and Raspberry Pi platforms. Covers embedded systems, networking, programming, and culminates in a hands-on capstone project.

https://www.coursera.org/specializations/iot

Introduction to the Internet of Things and Embedded Systems — UC Irvine (part of the specialization)

A focused standalone course introducing IoT fundamentals, components, networking, and embedded systems interactions. Suitable for auditing and provides foundational knowledge in IoT systems.

https://www.coursera.org/learn/iot

IoT Communications

Explores programming IoT devices with Arduino/Raspberry Pi, sensor technologies, communication protocols (e.g., MQTT), and foundational networking—all essential for IoT-focused education.

https://www.coursera.org/learn/iot-communications

Programming with Cloud IoT Platforms

Introduces cloud-based IoT services provided by major platforms (Samsung, Microsoft, Amazon, IBM, Google) and how to integrate them into IoT applications.

https://www.coursera.org/learn/cloud-iot-platform

IoT Systems and Industrial Applications with Design Thinking Specialization — LearnQuest

Focuses on designing IoT solutions for industrial automation using design-thinking methodologies, edge computing, IoT security, and robotics integration.

https://www.coursera.org/specializations/iot-systems-and-industrial-applications-with-design-thinking

etc.

The student must inform the teacher about the course taken or planned to be taken and clarify the results and prospects for crediting the learning outcomes obtained in non-formal/informal education.

Syllabus of the course

Is designed by teacher PhD, Associate Professor, Liubov Oleshchenko

Adopted by Computer Systems Software Department (protocol № 12, 11.02.2026)

Approved by the Faculty Board of Methodology (protocol № 7, 23.02.2026)