KHMELNYTSKYI NATIONAL UNIVERSITY

APPROVED

Dean of the Faculty of Information Technology

_______ Tetiana HOVORUSHCHENKO

Signature Name, LAST NAME

01 09 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

WORK PROGRAM OF THE EDUCATION COMPONENT

Software architecture and design

Field of study:	12 Information Technology
Specialty :	121 Software Engineering
Level of Higher education:	First (bachelor's) level
Educational and Professional Program:	Software Engineering
Course Load:	7 ECTS credits
Course code:	MPT.01
Language of instruction:	English
Educatial component:	Compulsory (Professional training)
Faculty:	Faculty Information Technology
Department:	Department Software Engineering

Form of Study	Year	Semester	Total Credits		Number of hours						Course project*	Course work	Semester Control Form	
					Classroom Hours					Independent work, incl. IRS				
			ECTS Loans	Hours	Tootal	Lecture	Laboratory work	Practical classes	Seminar classes				Passed	Exam
Etc	3	6	7	210	66	32	34			144	+			+

***Note**:* * In the academic discipline in the sixth semester, a course project is provided, the content and requirements for the implementation of which are regulated by the relevant methodological recommendations.

The work program is based on the educational and professional program "Software Engineering" in the specialty "Software Engineering"

Progarm's author _______ Yurii FORKUN _______
Signature Academic degree, academic title, Name, SURNAME

Approved at the meeting of the Department of ___ Software Engineering _____
Name

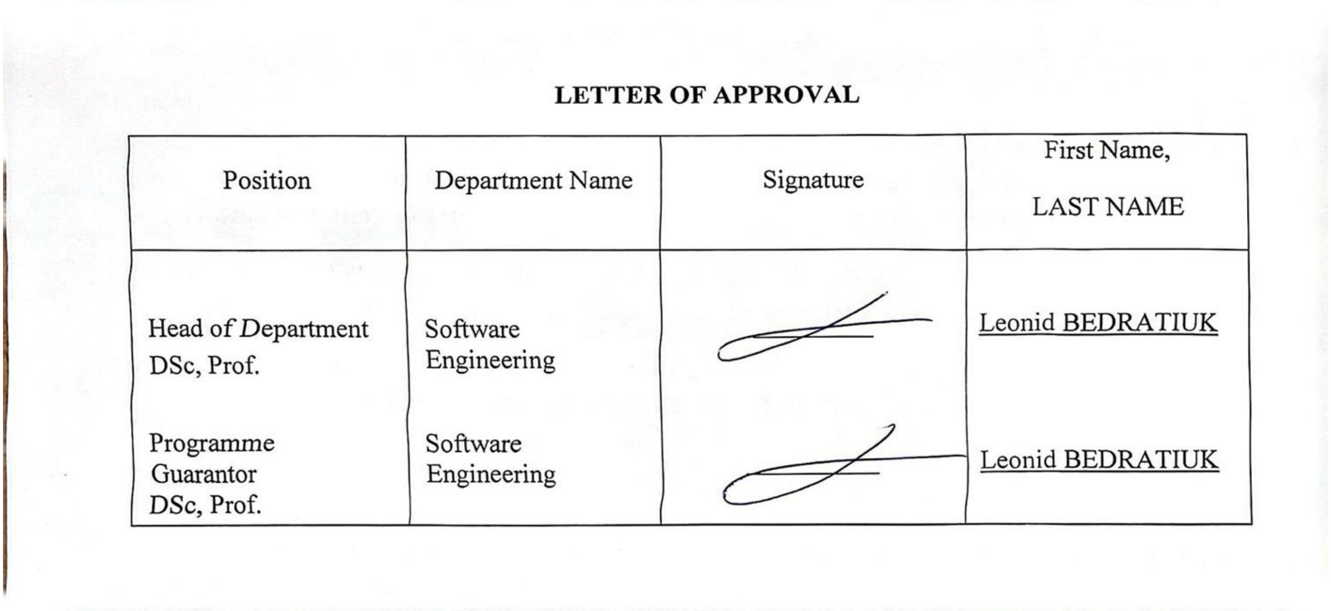
Minutes of dated No.1 August 28, 2025

Head of the Department of ___ Software Engineering _______ Leonid BEDRATYUK
Name Signature Name, SURNAME

The work programme was reviewed and approved by the Academic Council of the Faculty of Information Technologies

Chair of the Academic Council _______ Tetiana HOVORUSHCHENKO

Khmelnytskyi 2025



LETTER OF APPROVAL

Position	Department Name	Signature	First Name, LAST NAME
Head of *D*epartment DSc, Prof.	Software Engineering		<u>Leonid BEDRATIUK</u>
Programme Guarantor DSc, Prof.	Software Engineering		<u>Leonid BEDRATIUK</u>

3. Explanatory note

The discipline "Software Architecture and Design" is one of the professional disciplines and occupies a leading place in the training of applicants for the first (bachelor's) level of higher education, full-time (hereinafter referred to as full-time) form of higher education, who study under the educational and professional program "Software Engineering" within the specialty 121 "Software Engineering".

Prerequisites: web technologies (MPT10), human-machine interaction (MPT13).

Postrequisites: software design (MPT.08).

According to the educational program, the discipline contributes to ensuring:

competencies: Ability to solve complex, specialised tasks or practical problems in software engineering, characterised by complexity and uncertainty of conditions, using information technology theories and methods. (IC); Ability to participate in software design, including modelling (formal description) of its structure, behaviour, and operational processes (PC02); Ability to accumulate, process, and systematise professional knowledge regarding the creation and maintenance of software and recognise the importance of lifelong learning (PC10); Ability to implement phases and iterations of the life cycle of software systems and information technologies based on relevant software development models and approaches (PC11); Ability to execute the system integration process and apply standards and change management procedures to maintain the integrity, overall functionality, and reliability of the software. (PC12); Ability to reasonably choose and master the toolkit for software development and maintenance. (PC13).

program learning outcomes: To analyse, purposefully search for, and select the necessary information, reference resources, and knowledge for solving professional tasks, considering modern scientific and technical achievements (PLO01); To understand the software lifecycle's leading processes, phases, and iterations (PLO03); To conduct a pre-project survey of the subject area and system analysis of the design object (PLO10); To select initial data for design, guided by formal methods of requirement descriptions and modelling (PLO11); To apply effective software design approaches in practice (PLO12); To use instrumental software tools in practice for domain analysis, design, testing, visualisation, measurement, and software documentation (PLO14).

Purpose of the course. Based on the study of the theoretical foundations of modeling, development and design based on software templates and frameworks, as well as the study of the basic standards, requirements and ready-made structures used in the development and design of large software applications, students should acquire practical skills in creating architecture and detailed design of software products using modern methods and tools.

Subject of the course.. Theory and practice of application of basic methods and tools for the development and design of software architecture.

Subject of the course. To provide students with knowledge and practical skills in the basics of software architecture design and development.

Learning outcomes. A student who successfully completes the course must understand the essential characteristics of the architecture of software applications and approaches to their analysis and design, the main types of architectures and their features, software design tasks; use the UML modeling language for designing software applications; use standard tools and notations to document the architecture and interface of the software; master the tools of software design and methods of structural and object-oriented software design; be able to use architectural design patterns and apply them when creating software applications; create software using typical architectural styles and solutions, taking into account the peculiarities of the application; design enterprise software using the typical architecture of corporate systems and platforms.

4. Structure of the course credits

Name of the topic (topic)	Number of hours allotted to:		
	Lecture	Laboratory work	SRS
Topic 1. Introduction to Software Architecture	2		12
Topic 2. Models, wireframes and design samples	4	4	12
Topic 3. Types of architectures and their models	2	4	12
Topic 4. Architecture: Notation, Standards, and Design Tools	2	4	12
Topic 5. Architectural Patterns and Styles	2	4	12
Topic 6. Quality control when choosing an architecture	2		12
Topic 7. Introduction to detailed design	4	4	12
Topic 8. Sequence diagrams and data flow diagrams in software design	2	4	12
Topic 9. Specification of algorithms, classes and functions	4		12
Topic 10. Design samples	4	4	12
Topic 11. Standard template libraries	4	4	12
Topic 12. The Impact of Detailed Design on the Project	2	2	12
Together:	**16**	**34**	**144**

5. Program of the academic course

5.1. Content of the lecture course

№ Lecture	List of lecture topics, their annotations	Number of hours
Topic 1. Introduction to Software Architecture		2
1	Introduction to Software Architecture. Overview of the development technology. The concept of software architecture. Goals of choosing architecture. Decomposition [1]; [2]; [3]; [4]; [5]	2
Topic 2. Models, wireframes and design samples		4
2	Models, frames and design samples. Use of models. Unified Modeling Language (UML). Frameworks. [1]; [2]; [3]	2
3	Models, frames and design samples. Classification of architectures. Design samples. Components. [1]; [4]	2
Topic 3. Types of architectures and their models		
4	Types of architectures and their models. Architectures based on data flows. Independent components. Virtual machines. Repository architectures. Level architectures. Applications with mixed architecture. Architecture selection procedure [1]; [2] ; [4]; [5]	2
Topic 4. Architecture: Notation, Standards, and Design Tools		
5	Architecture: Notation, Standards, and Tools. Notation. Tools. IEEE/ANSI standard for project description [1]; [2] ; [3]; [4]	2
Topic 5. Architectural Patterns and Styles		2
6	Architectural patterns and styles. Main architectural styles. A combination of architectural styles. Client/server architecture. Component architecture. Domain-based design. Multi-layered architecture. Object-oriented architecture. Architecture based on the notification bus. Service-oriented architecture. [1]; [2]; [3] ; [4]; [6]	2
Topic 6. Quality control when choosing an architecture		2
7	Quality control when choosing an architecture. Quality and choice of architecture. Choice of alternative architectures. Validate architecture with use cases. Inspection of architecture selection. Impact of Architecture Choice on SPMP. [1]; [2]; [3]; [6]	2
Topic 7. Introduction to detailed design		4
8	Introduction to detailed design. The concept of detailed design. Correlation of use cases, architecture and detailed design. A typical diagram of the detailed design process. [1]; [2] ; [3] ; [4]; [5]	2
9	Introduction to detailed design. USDP design – requirements, analysis, design, implementation, testing. Design and interfaces. Reusable items. [1]; [2] ; [3] ; [5]	2
Topic 8. Sequence diagrams and data flow diagrams in software design		2
10	Sequence diagrams and data flow diagrams in detailed design. Detailed sequence diagrams. Detailed data flow diagrams [1]; [2] ; [3] ; [4]; [6]	2
Topic 9. Specification of algorithms, classes and functions		4
11	Specification of algorithms, classes and functions. Class invariants. Invariants, prerequisites and postconditions of functions. Flowcharts. Pseudocode. Use of flowcharts and pseudocode [1]; [2]; [3]; [4]	2
12	Specification of algorithms, classes and functions. Class invariants. Invariants, prerequisites and postconditions of functions. Flowcharts. Pseudocode. Use of flowcharts and pseudocode [1]; [2]; [3]; [4];][5]; [6]	2
Topic 10. Design samples		4
13	Design samples. Detailed design techniques. Creation design samples. Structural design samples. Design patterns based on application behavior [1]; [2]; [3]; [4]; [6]	2
14	Design samples. Patterns of behavior. Anti-patterns of design [3]; [4]; [6]	2
Topic 11. Standard template libraries		2

15	Standard template libraries. Standards, notation, and design tools. C++ Standard Template Library (STL). [3]; [4]; [6]	2
Topic 12. The Impact of Detailed Design on the Project		2
16	The impact of detailed design on the project. Estimation of the scope of work using detailed design. Quality and metrics in detailed design. Inspection for detailed design [1]; [2] ; [3] ; [4]; [5]	2
	Generally	32

5.2. Content of laboratory works

№	List of topics of laboratory work, their annotations	Number of hours
1	Lab Work No. 1 Evaluation and Selection of Software Components	4
2	Lab No. 2 Using different types of software architectures. Design patterns	4
3	Lab No. 3 Working Mixed-Architecture Applications	4
4	Laboratory work No. 4 Use of standards and tools in the development of software architecture	4
5	Laboratory work No. 5 Initial stage of design. Subject Area Research	4
6	Lab No.: Software System Architecture Development: Choosing the Type of Architecture and Decomposition Description	4
7	Laboratory work No. 7 Development of software system architecture: description of dependencies and description of the interface	4
8	Laboratory work No. 8 Detailed design of the software system: modules and data	4
9	Generalizing lesson	2
	Generally	**34**

5.3 The content of independent work of *full-time* students

Independent work of students of all forms of education consists in the systematic study of program material from appropriate sources of information, preparation for the implementation and defense of laboratory work, testing. In addition, students have at their disposal a page of the academic discipline in the Modular Learning Environment, where the Work Program of the discipline and the necessary documents for its educational and methodological support are posted.

Number of the week	Type of independent work	A few hundred hours
1	Elaboration of lecture material. Preparation for laboratory work No. 1.	6
2	Elaboration of lecture material. Processing the results of laboratory work No. 1.	6
3	Elaboration of lecture material. Preparation for laboratory work No. 2.	6
4	Elaboration of lecture material. Elaboration of the results of laboratory work No. 2.	6
5	Elaboration of lecture material. Preparation for laboratory work No. 3.	6
6	Elaboration of lecture material. Elaboration of the results of laboratory work No. 3. Preparation for test control.	6
7	Elaboration of lecture material. Preparation for laboratory work No. 4.	6
8	Elaboration of lecture material. Elaboration of the results of laboratory work No. 4.	8
9	Elaboration of lecture material. Preparation for laboratory work No. 5.	6
10	Elaboration of lecture material. Elaboration of the results of laboratory work No. 5.	6
11	Elaboration of lecture material. Preparation for laboratory work No. 6.	6
12	Elaboration of lecture material. Elaboration of the results of laboratory work No. 6. Preparation for test control.	6
13	Elaboration of lecture material. Preparation for laboratory work No. 7.	6
14	Elaboration of lecture material. Elaboration of the results of laboratory work No. 7.	6
15	Elaboration of lecture material. Preparation for laboratory work No. 8.	6
16	Elaboration of lecture material. Elaboration of the results of laboratory work No. 8.	8
	Overall for the third semester	**100**

6. Technologies and teaching methods

The learning process in the discipline is based on the use of traditional and modern methods. In particular, lectures are conducted by verbal methods, and laboratory classes are conducted by practical and visual methods using information technologies and modern means of their implementation and are aimed at acquiring practical skills in software architecture development and design by students and have the goal of mastering the basic concepts of software architecture and the use of tools for their design and implementation.

7. Methods of assessment

Current control is carried out during classroom laboratory classes, as well as on the days of control measures established by the work program and schedule of the educational process, including the use of the Modular Learning Environment. The following methods of current control are used:

– oral interview before admission to the laboratory lesson;

– evaluation of the results of laboratory work defense

– test control of the assimilation of theoretical material on the topic.

When deriving the final semester grade, the results of both current control and final control, which is carried out from all the material of the discipline according to tickets previously developed and approved at the meeting of the department, are taken into account. A higher education applicant who has scored in any type of educational work the amount of points lower than 60 percent of the maximum score is not allowed to semester control until he completes the amount of work provided for by the Work Program. A higher education applicant who has scored a positive weighted average score (60 percent or more of the maximum score) in all types of current control and has not passed the exam is considered to have academic debt. Elimination of academic debt from semester control is carried out during the examination session or according to the schedule established by the dean's office in accordance with the "Regulations on control and evaluation of the learning outcomes of higher education applicants at KhNU".

8. Course policy

The policy of the academic discipline is generally determined by the system of requirements for the applicant for higher education, which are provided for by the current provisions of the University on the organization and educational and methodological support of the educational process. In particular, passing a safety briefing; Attendance at discipline classes is mandatory. For objective reasons (confirmed by documents), theoretical training in agreement with the lecturer can take place online. Successful mastery of the discipline and the formation of professional competencies and program learning outcomes implies the need to prepare for a laboratory lesson (study of theoretical material on the topic of work, preliminary preparation of the work protocol, preparation for an oral interview for admission to the lesson (given in the Methodological Recommendations for Laboratory Classes)), actively work in the classroom, prepare a high-quality report (in accordance with the topic, task and purpose), defend the results of the work performed, participate in discussions on the constructive decisions made when performing laboratory work by applicants, etc.

Applicants for higher education must comply with the established deadlines for the implementation of all types of educational work in accordance with the work program of the academic discipline. The term of defense of laboratory work is considered timely if the student defended it at the next lesson after completing the work). The student is obliged to work out the missed laboratory lesson within the deadline set by the teacher, but no later than two weeks before the end of theoretical classes in the semester.

The student's assimilation of theoretical material in the discipline is evaluated according to the results of evaluation of laboratory work, testing and performance of tests and test controls. The implementation of the course project ends with its submission for verification, the content and requirements for the implementation of which are regulated by the relevant methodological recommendations.

In case of violation of the academic integrity policy in any types of academic work, the applicant for higher education receives an unsatisfactory grade and must repeat the tasks on the relevant topic (type of work) provided for by the work program. Any forms of violation of academic integrity ***are not allowed.***

Within the framework of the study of the academic discipline, applicants for higher education are provided with the recognition and enrollment of learning outcomes acquired through non-formal education, which are posted on accessible platforms that contribute to the formation of competencies and deepening of learning outcomes determined by the work program of the discipline, or provide the study of the relevant topic and/or the type of work on the program of the academic discipline (more details in the Regulations on the procedure for recognizing and enrolling the learning outcomes of higher education applicants at KhNU).

9. Assessment of students' learning outcomes during the semester

Assessment of academic achievements of a higher education applicant is carried out in accordance with the "Regulations on Control and Evaluation of Learning Outcomes of Higher Education Applicants at KhNU". During the current assessment of the work performed by the applicant from each structural unit and the results obtained by him, the teacher assigns him a certain number of points from the Work Program for this type of work. At the same time, each structural unit of educational work can be credited if the applicant has scored at least 60 percent (the minimum level for a positive assessment) of the maximum possible amount of points assigned to the structural unit.

When evaluating the learning outcomes of higher education applicants in any type of educational work (structural unit), it is recommended to use the following generalized criteria:

Table – Assessment Criteria for Student Learning Outcomes

Grade and Level of Achievement of Intended Learning Outcomes and Competences	**General Description of Assessment Criteria**
Excellent (high)	The student has deeply and fully mastered the course content, confidently navigates it, and skilfully uses the conceptual framework; demonstrates the ability to connect theory with practice, solve practical problems, and clearly express and justify their reasoning. An excellent grade implies a logical presentation of the answer in the language of instruction (oral or written), high-quality formatting of the work, and proficiency in using specialised tools, instruments, or application software. The student demonstrates confidence when answering reformulated questions, is capable of making detailed and summarised conclusions, and shows practical skills in solving professional tasks. The answer may contain two or three minor inaccuracies.
Good (*Average*)	The student has shown full understanding of the course content, possesses the conceptual framework, and navigates the material well; applies theoretical knowledge consciously to solve practical tasks. The answer is generally well-articulated, although some minor inaccuracies or vague formulations of rules or principles may occur. The student's answer is based on independent thinking. Two or three minor mistakes are acceptable.
Satisfactory (sufficient)	The student demonstrates knowledge of the basic course material sufficient for continued learning and practical activity in the profession; is able to complete the practical tasks foreseen by the programme. The answer is usually based on reproductive thinking. The student has limited knowledge of the structure of the discipline, makes inaccuracies and significant errors in the answer, and hesitates when answering reformulated questions. Nevertheless, they possess basic skills to complete simple practical tasks that meet the minimum assessment criteria and, under the lecturer's guidance, can correct their mistakes.
Unsatisfactory (insufficient)	The student demonstrates knowledge of the basic course material sufficient for continued learning and practical activity in the profession; is able to complete the practical tasks foreseen by the programme. The answer is usually based on reproductive thinking. The student has limited knowledge of the structure of the discipline, makes inaccuracies and significant errors in the answer, and hesitates when answering reformulated questions. Nevertheless, they possess basic skills to complete simple practical tasks that meet the minimum assessment criteria and, under the lecturer's guidance, can correct their mistakes.

Based on the results of the current control, the final semester grade is assigned.

Structuring of the Course by Types of Academic Work and Assessment of Student Learning Outcomes

In-Class Work								***Assessment Activities***		***Semester Final Assessment***	
Laboratory Work №:								Test Control:		Exam	Total points
1	2	3	4	5	6	7	8	T1	T2		
Number of points per type of academic work (min–max)											
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	6-10	6-10	**24-40**	**60-100***
24-60								**12-20**		**24-40**	

Notes: If the number of points earned for any type of academic work in the course is below the established minimum, the student receives a failing grade and must retake the work within the deadline set by the lecturer (or dean). The institutional grade is determined in accordance with the table "**Correspondence between the Institutional Grading Scale and the ECTS Grading Scale**".

Laboratory work performed and designed in accordance with the requirements established by the Methodological Recommendations is comprehensively evaluated by the teacher during its defense, taking into account the following criteria: independence and correctness of performance; completeness of the answer and knowledge of the work methodology; the presence of screenshots, diagrams and compliance with the requirements for the design of figures, etc.

The result of the performance and defense of each laboratory work by the applicant for higher education is evaluated in accordance with the table of Criteria for evaluating the educational achievements of the applicant for higher education.

If the applicant reveals a level of knowledge lower than 60 percent of the maximum score established by the Work Program for each structural unit, laboratory work ***is not credited to him,*** and in order to defend it, he must study in more detail the material on the topic of the work, the method of its implementation, correct gross mistakes and re-defend it at the time appointed by the teacher.

Assessment of Test-Based Control Results

Each of the tests provided for by the Work Program consists of 10 test tasks, each of which is equivalent.

According to the table of structuring the types of work for test control, the applicant, depending on the number of correct answers, can receive from 6 to 10 points.

Distribution of points depending on correct answers to test questions

Number of correct answers	0-5	6	7	8	9	10
Percentage of correct answers	**0-59**	**60**	**70**	**80**	**90**	**100**
Number of points received	0	6	7	8	9	10

30-40 minutes are allotted for testing. The student is tested online in the Modular Learning Environment. Also, the student can take the test in writing, writing down the correct answers in the answer card.

For each individual type of tasks of the final semester control, the criteria for assessing the educational achievements of a higher education applicant, given above (**Table – Criteria for evaluating the educational achievements of a higher education applicant),** are applied.

The final semester grade on the institutional scale and the ECTS scale is determined in an automated mode after the teacher enters the assessment results in points for all types of educational work into the electronic journal. The ratios of the institutional evaluation scale and the ECTS evaluation scale are given below in the table "Ratio".

Evaluation of the results of the final semester control (exam)

The educational program provides for the final semester control in the discipline in the form of an exam, the task of which is a systematic and objective assessment of both theoretical and practical training of the applicant in the academic discipline. The exam is taken according to tickets previously developed and approved at the meeting of the department. In accordance with this, the exam ticket offers a combination of questions of both theoretical (including in test form) and practical.

Table – Assessment of Final Semester Examination Results *for full-time students (40 points allocated for final control)*

Types of tasks	**For each individual type of task**		
	Minimum (sufficient) score (satisfactory)	**Potential positive scores* (GPA) (good)**	**Maximum (high) score (excellent)**
Theoretical question No. 1	3	4	5
Theoretical question No. 2	3	4	5
Practical task (task)	18	24	30
Together:	**24**		**40**

The semester exam is set if the total amount of points scored by the student in the discipline according to the results of the current control is in the range from 60 to 100 points. At the same time, according to the institutional scale, the grade is "excellent/good/satisfactory", and according to the ECTS scale - the letter designation of the grade, which corresponds to the sum of points scored by the student in accordance with the Ratio table.

Table – Correspondence between the Institutional Grading Scale and the ECTS Grading Scale

ECTS Grade	**Rating Scale of Points**	**Institutional Grade(Level of Achievement of the Intended Learning Outcomes in the Course)**	
		Passed	**Exam/ Graded Credit**
A	90-100	Pass	**Excellent** – a high level of achievement of the intended learning outcomes in the course, indicating the learner's full readiness for further study and/or professional activity in the field.
B	83-89		**Good** – an average (maximally sufficient) level of achievement of the intended learning outcomes in the course and readiness for further study and/or professional activity in the field.
C	73-82		
D	66-72		**Satisfactory** – the student has demonstrated a minimally sufficient level of achievement of the learning outcomes required for further study and/or professional activity in the field.
E	60-65		
FX	40-59	Fail	**Fail** – several intended learning outcomes in the course have not been achieved. The level of acquired learning outcomes is insufficient for further study and/or professional activity in the field.
F	0-39		**Fail** – no learning outcomes have been achieved.

10. Self-assessment questions on learning outcomes

1. The concept of system development.
2. The concept of software.
3. The concept of creating architecture.
4. Goals of choosing architecture.
5. The concept of decomposition.
6. The concept of detailed design.
7. Use of models.
8. Frameworks.
9. Classification of architectures.
10. Design samples.
11. Components.
12. Types of architectures and their models.
13. Architectures are based on data flows.
14. Independent components.
15. Virtual machines.
16. Repository architectures
17. Level architectures.
18. Client-server architecture.
19. Architecture is domain-based.
20. Architecture selection procedure.
21. Tools for choosing architecture.
22. Quality control when choosing an architecture.
23. Validate architecture with use cases.
24. Inspection of architecture selection.
25. The concept of detailed design.
26. The ratio of options for using architecture and detailed design.
27. A typical detailed design scheme.
28. Design and interfaces.
29. Reusable components.
30. Detailed sequence diagrams.
31. Detailed data flow diagrams.
32. Specification of classes and functions.
33. Class invariants.
34. Invariants, prerequisites and postconditions.
35. Specification of algorithms.
36. Flowcharts.
37. Pseudocode.
38. Comparison of the use of flowchart and pseudocode.
39. Design samples: detailed design techniques.
40. Creation design samples.
41. Structural design samples.
42. Design samples are based on the behavior of applications.
43. What is a design pattern?
44. What types of design patterns do you know?
45. Design template "Strategy".
46. Design template "Builder".
47. Design template "Factory", "Abstract Factory".
48. Adapter design pattern.
49. Bridge design template.
50. Command design pattern.
51. Design template "Linker".
52. Iterator design pattern.
53. Design template "Mediator".
54. Design template "Chain of responsibility".
55. The Explorer design pattern.
56. The "Intermediary" design pattern.
57. Design template "Loner".
58. Design pattern "Status".
59. Design pattern "Sample".
60. What is Software Design Quality Analysis?
61. Basic Software Design Assessments.
62. What notations and design support tools are known?
63. Methods for analyzing compromise architectural solutions.
64. The main stages of the method of analysis of compromise architectural solutions.
65. A method for analyzing the cost and efficiency of architectural solutions and software applications.

11. Educational and methodological support

The educational process in the discipline is fully and in sufficient quantity provided with the necessary educational and methodological literature. In particular, the teachers of the department prepared and published the following works:

1. Forkun Y.V. Architecture and Software Design: Methodical Instructions for Laboratory Work for Students for Students of the Specialty 121 Software Engineering. Khmelnytskyi: KhNU, 2023.

2. Forkun Y.V. Architecture and Software Design: Methodical Instructions for the Implementation of the Course Project for Students of the Specialty 121 Software Engineering. Khmelnytskyi: KhNU, 2021.

12. Material and technical and software of the discipline (if necessary)

Information and computer support: PC, tablet, smartphone or other mobile device, projector. Software: Microsoft Office programs or similar, Internet access, work with presentations, Visual Studio or analogues, tools for working with UML diagrams.

The study of the discipline does not require the use of special application software, except for commonly used programs and operating systems.

13. Recommended literature

Primary

1. Synopsis of lectures on the discipline "Software Architecture and Design" for applicants for higher education of the first (bachelor's) level under the educational and professional program "Software Engineering" in the specialty 121 – "Software Engineering" / Incl. V.V. Zavgorodniy, K.M. Yalova. – Kamianske: DDTU, 2019.– 144 p.

2. Martin R. Pure Architecture. Kharkiv: Fabula, 2019. 416 c.

3. Freeman E. Head First. Design patterns. Kharkiv: Fabula, 2020. 672 c.

4. Richards M. Fundamentals of Software Architecture: An Engineering Approach. Sebastopol, California: O'Reilly Media. 1st edition, 2020. 419 p.

5. Lanciaux R. Modern Front-end Architecture: Optimize Your Front-end Development with Components, Storybook, and Mise en Place Philosophy. New York: Apress, 2021. 144 p.

6. Frighi V. Smart Architecture. A Sustainable Approach for Transparent Building ComponentsDesign. Berlin: Springer – 1st edition, 2022. – 293 p.

7. Kudryavtsev V. V., Forkun Y. V. Analysis and application of methods of optimization of performance and fault tolerance of software products. Sciences. Pr. scientific conference "APKN-89 2021". Khmelnytskyi KhNU. – 2021. – p.338-339

8. Forkun, Y., Forkun, I., Yashina, O., & Pravorska, N. (2023). Architectural methods for optimizing the performance and fault tolerance of software applications. measuring and computing devices in technological processes, (2), 196–201. https://doi.org/10.31891/2219-9365-2023-74-27

Supplementary

1. Borodkina I.L., Borodkin G.O. Software Engineering: Manual. manual of NUBiP. Kyiv: Center for Educational Years, 2020. 204 p. (in Russian).

2. Lanciaux R. Modern Front-end Architecture: Optimize Your Front-end Development with Components, Storybook, and Mise en Place Philosophy. Ryan Lanciaux. New York: Apress, 2021. 144 p.

3. Frighi V. Smart Architecture A Sustainable Approach for Transparent Building ComponentsDesign. Berlin: Springer 1st edition, 2022. 293 p.

4. Robert Martin. Pure architecture. Kharkiv: Plot. 2019. 368 p. (in Russian). ISBN : 978-617-09-5286-8

5. Robert Martin. Clean code. Kharkiv: Plot. 2019. 416 p. (in Russian). ISBN : 978-617-09-5285-1

6. Oleksiy Vasiliev. Programming in Python. Ternopil: EDUCATIONAL BOOK BOGDAN. 2019. – 504 p. ISBN: 9789661056113

7. Elizabeth Robson , Eric Freeman. Head First. Design patterns. Kharkiv: Plot. 2020. 672 p. (in Russian). ISBN : 978-617-09-6159-4

8. Elizabeth Robson , Eric Freeman. Head First. JavaScript programming. Kharkiv: Plot. 2022. 672 p. (in Russian). ISBN: 978-617-522-047-4

9. Paul Berry. Head First. Python. Kharkiv: Plot. 2021. 624 p. (in Russian). ISBN : 978-617-522-019-1

10. Rob Cole. Brilliant Agile. Kharkiv: Plot. 2020. 192 p. (in Russian). ISBN : 978-617-09-6381-9

11. Bert Bates , Katie Sierra. Head First. Java. Kharkiv: Plot. 2022. 720 p. ISBN : 978-617-522-033-7

12. Yuriy Ramsky, Vasyl Oleksyuk, Anatoly. Administration of computer networks and systems. Ternopil: EDUCATIONAL BOOK BOGDAN. 2020. 196 p. ISBN : 9789661015615

13. Amy Webb. The Big Nine. How IT giants and their smart machines can change humanity. Kharkiv: Vivat. – 2020. – 352 p. ISBN : 9789669822185

14. Denis Kaplunov. The kings of social networks. Kyiv: BookChef. 2022. – 432 p. ISBN: 9786175480922

14. Information resources

1. Modular learning environment. URL: https://msn.khmnu.edu.ua/course/view.php?id=4908
2. Electronic library of the university. URL: http://lib.khmnu.edu.ua/asp/php_f/p1age_lib.php
3. Repository of KhNU. URL: http://elar.khmnu.edu.ua/home/

Software architecture and design

Type of Educational Component	Compulsory
Level of Higher Education	First (Bachelor's) Level
Language of instruction	Ukrainian
Semester	Sixth
Number of ECTS credits assigned	5,0
Forms of education for which the discipline is taught	Full-time (daytime)

Learning outcomes. A student who successfully completes the course must understand the essential characteristics of the architecture of software applications and approaches to their analysis and design, the main types of architectures and their features, software design tasks; use the UML modeling language for designing software applications; use standard tools and notations to document the architecture and interface of the software; master the tools of software design and methods of structural and object-oriented software design; be able to use architectural design patterns and apply them when creating software applications; create software using typical architectural styles and solutions, taking into account the peculiarities of the application; design enterprise software using the typical architecture of corporate systems and platforms.

Course content. Introduction to Software Architecture. Models, Frameworks, and Design Samples. Types of Architectures and Their Models. Architecture: Notation, Standards, and Design Tools. Architectural Patterns and Styles. Quality Control in Architecture Selection. Introduction to Detailed Design. Sequence Diagrams and Data Flow Diagrams in Software Design. Specification of Algorithms, Classes, and Functions. Design Samples. Standard Template Libraries. The Impact of Detailed Design for the project

Prerequisites: web technologies (OPP10), human-machine interaction (OPP13).

Corrections: software design (OPP.08).

Planned Learning Activities.: The minimum volume of classes in one ECTS credit of the academic discipline for the first (bachelor's) level of higher education in full-time education is 10 hours per 1 ECTS credit.

Forms (methods) of Instruction: lectures (using visualization methods, problem-based and interactive learning, motivational techniques**,** information and communication technologies)**;** laboratory classes (using methods of information technology and modern integrated programming environments, master classes); independent work (processing of theoretical material, preparation for the implementation and defense of laboratory work, current and final control) using information and computer technologies and distance learning technologies.

Assessment Methods: oral survey before admission to the laboratory lesson; evaluation of the results of laboratory work defense; test control of the assimilation of theoretical and practical material.

Form of Final Assessment: exam – 6th semester.

Learning Resources:

1. Martin R. Pure Architecture. Kharkiv: Fabula, 2019. 416 c.
2. Freeman E. Head First. Design patterns. Kharkiv: Fabula, 2020. 672 c.
3. Richards M. Fundamentals of Software Architecture: An Engineering Approach. Sebastopol, California: O'Reilly Media. 1st edition, 2020. 419 p.
4. Lanciaux R. Modern Front-end Architecture: Optimize Your Front-end Development with Components, Storybook, and Mise en Place Philosophy. New York: Apress, 2021. 144 p.
5. Modular learning environment. Access to resource: https://msn.khmnu.edu.ua/course/view.php?id=4908
6. Electronic library of KhNU. Access to the resource: http://library.khmnu.edu.ua/

Teachers: PhD, Associate Professor Forkun Y.