KHMELNYTSKYI NATIONAL UNIVERSITY

APPROVED
Dean of IT Faculty
_____ HOVORUSHCHENKO T.
"09" _____ 2025.

WORKING PROGRAMME OF THE EDUCATIONAL COMPONENT
Software Modelling and Evaluation

Field of Study: 12 – Information Technology
Specialty: 121 – Software Engineering
Level of Higher Education: First (Bachelor's) Level
Educational and Professional Programme: Software Engineering
Course Load: 5 ECTS credits **Course Code:** CPT.09
Language of Instruction: English
Status of the Educational Component: Compulsory (General Training)
Faculty: Faculty of Information Technology
Department: Department of Software Engineering

Form of Study	Year	Semester	Total Credits		Number of hours						Course project	Coursework	Semester control form	
					Contact Hours					Independent Work (incl. Individual Tasks)				
			ECTS credits	hours	Total	Lectures	Laboratory works	Practical classes	Seminar classes				pass/ fail test	Exam
D	3	5	5	150	66	32	34			84				+

The working programme is based on the Educational and Professional Programme "Software Engineering" within the specialty 121 "Software Engineering".

Program's author _____ O.G.Onyshko

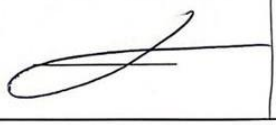
Approved at the meeting of the Department of Software Engineering
Minutes No. 1 dated August 30, 2025
Head of the Department _____ L.P. Bedratyuk

The working programme was reviewed and approved by the Academic Council of the Faculty of Information Technology

Chair of the Academic Council _____ Tetiana HOVORUSHCHENKO

Khmelnytskyi 2025

LETTER OF APPROVAL

Position	Department Name	Signature	First Name, LAST NAME
Head of Department DSc, Prof.	Software Engineering		Leonid BEDRATIUK
Programme Guarantor DSc, Prof.	Software Engineering		Leonid BEDRATIUK

SOFTWARE MODELLING AND EVALUATION

Type of Educational Component	Compulsory
Level of Higher Education	First (Bachelor's) Level
Language of Instruction	English
Semester	Fifth
Number of ECTS Credits Assigned	5
Forms of Study the Course is Designed For	Full-time

Learning Outcomes. According to the Standard of Higher Education and Educational Program, the discipline should provide:
Competencies: the ability to identify, classify and formulate requirements for software; the ability to formulate and provide requirements for software quality in accordance with the requirements of the customer, technical specifications and standards; the ability to comply with specifications, standards, rules and recommendations in the professional field when implementing life cycle processes; the ability to accumulate, process and systematize professional knowledge on the creation and maintenance of software and recognition of the importance of lifelong learning; the ability to implement phases and iterations of the life cycle of software systems and information technologies based on appropriate models and approaches to software development; the ability to carry out the system integration process, apply standards and change management procedures to maintain the integrity, overall functionality and reliability of the software
Program learning outcomes: analyze, purposefully search and select the information and reference resources and knowledge necessary for solving professional tasks, taking into account modern achievements of science and technology; know the main processes, phases and iterations of the software life cycle; know and be able to use methods and tools for collecting, formulating and analyzing software requirements; know approaches to assessing and ensuring software quality

Course Content. General approaches to modeling, Principles of modeling. Fundamentals of structural modeling of software, Fundamentals of modeling of software behavior. Fundamentals of event modeling. Software architecture modeling.
Planned Learning Activities. The minimum amount of classroom-based learning activities in one ECTS credit for a course at the first (Bachelor's) level of higher education in full-time study mode is 10 hours per 1 ECTS credit.

Forms (Methods) of Instruction: Lectures (using problem-based learning and visualisation methods), Laboratory works, Independent work

Assessment Methods: Laboratory work defence, Testing

Form of Final Assessment: Exam

Learning Resources:

1. Sommerville, I. Software Engineering (10th ed.). Pearson Education, 2021. – 816 p.
2. Pohl, K., Rupp, C. Requirements Engineering: Fundamentals, Principles, and Techniques. Springer, 2022. – 400 p.
3. Tavana, M., Pick, K. W. Managing Requirements Knowledge. Springer, 2022. – 300 p.
4. Введення в програмну інженерію і управління життєвим циклом програмного забезпечення Guide to Software Engineering Base of Knowledge (SWEBOK): Пер. з англ. С.Орлик Retryeved from:sorlik.blogspot.com/
   Модульне середовище для навчання MOODLE. Доступ до ресурсу: https://msn.khmnu.edu.ua/

Lecturer: Candidate of Pedagogical Sciences, Associate Professor Onyshko O.G.

3. EXPLANATORY NOTE

The course "Software Modelling and Evaluation" is one of the general training courses and occupies a leading place in the training of students of the first (Bachelor's) level of higher education, full-time mode of study (hereinafter – full-time), who study under the Educational and Professional Programme "Software Engineering" within the specialty F2 "Software Engineering".

Prerequisites – CPT.05 Software Engineering Basics CPT.14 Software Requirement Analysis and Quality CGT.15 Algorithms and Data Structures

Postrequisites – CPT.09 Software Construction CPT.18 Qualification Work

In accordance with the educational programme, the course contributes to the development of: **competences:**. IC**.** 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. PC1. Ability to identify, classify, and formulate software requirements. PC2. Ability to participate in software design, including modelling (formal description) of its structure, behaviour, and operational processes. PC3. Ability to develop architectures, modules, and components of software systems. PC5. Ability to adhere to specifications, standards, rules, and recommendations in the professional field during the implementation of lifecycle processes. PC10. Ability to accumulate, process, and systematise professional knowledge regarding the creation and maintenance of software and recognise the importance of lifelong learning. PC11. Ability to implement phases and iterations of the life cycle of software systems and information technologies based on relevant software development models and approaches. PC12. 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.

programme learning outcomes:. PLO5 To understand and apply relevant mathematical concepts, domain and system methods, object-oriented analysis, and mathematical modelling for software development. PLO6 To select and utilise a software development methodology appropriate for the task. PLO10 To conduct a pre-project survey of the subject area and system analysis of the design object. PLO11 To select initial data for design, guided by formal methods of requirement descriptions and modelling. PLO13 Know and apply methods for developing algorithms, designing software, and data and knowledge structures. PLO14 To use instrumental software tools in practice for domain analysis, design, testing, visualisation, measurement, and software documentation. PLO17 To be skilled in applying methods of component software development. PLO19 To know and apply methods for software verification and validation. PLO20 To know approaches to software quality evaluation and assurance.

Purpose of the course. consists in providing theoretical and practical training of students, which should ensure that students obtain basic knowledge in the field of modern design technologies, software requirements engineering, acquire practical skills in the implementation of software systems, the basics of modeling and analysis of software systems, development analysis, specification and requirements management.

Subject of the course. The subject of the discipline "Software Modeling and Evaluation" covers theoretical knowledge, tasks, methods and requirements for software, its design and construction processes.

Course objectives. The main objectives of studying the discipline "Software Modeling and Evaluation" are knowledge about the development and analysis of requirements that are put forward for a software product. Requirements are classified, requirements properties are analyzed, methodologies, standards, and notations for working with requirements are considered. The

components of requirements analysis are analyzed: identification, specification and documentation, verification. The role of models, tools, and requirements management processes is considered.
Learning Outcomes. Upon successful completion of the course, the student should be able to: *formalise* computational and logical problems; *represent* algorithms for computational and logical problems using pseudocode, flowcharts, or programming languages; *analyse* the efficiency of algorithms in terms of time and space complexity; *use* basic data structures (arrays, sets, queues, stacks, lists, trees, graphs); *apply* methods for constructing complex data structures based on simple ones; *perform* data search using linear, binary, and pattern matching methods; *implement* text search using classical algorithms and regular expressions; *select* the optimal data structure and algorithm for solving a problem with regard to performance; *use* standard libraries and tools for data processing in modern programming languages; *carry out* experimental analysis and visualisation of algorithm performance, and *choose* the optimal data structure and algorithm to solve a given problem.

4. STRUCTURE OF THE COURSE CREDITS

Topic Title	Number of hours allocated to:		
	Lectures	Lab work	Independent work
Topic 1. Introduction. Characteristics of approaches to software modeling	4	6	4
Topic 2 Basic concepts of structural software modeling	8	6	20
Topic 3. Basic concepts for modeling software behavior.	12	8	20
Topic 4. Fundamentals of event modeling in software	4	6	20
Topic 5. Software architecture modeling	4	8	20
Total for the semester	32	34	84

5.1. CONTENT OF THE LECTURE COURSE

Lecture No.	List of Lecture Topics and Annotations	Hours
Topic 1. Algorithms and their analysis		
1	Overview of software development techniques. Generating correct code with minimal volume. Software modeling. The concept of a system model. Informal and formal modeling Ref.: [1] pp. 30–40; [2] pp. 6–9; [3] pp. 18–28	4
Topic 2. Sorting algorithms		
2	Classes. Class requirements. Attributes. Attribute requirements. Creating attributes and operations. Class responsibilities. Stereotypes, assigned values, constraints. Modeling comments. Classifier. Extended classes and their properties. Types of classifiers. Template classes. Instance. Types of instances. Automaton. Active and passive elements .Ref.: [1] pp. 41–42; [2] pp. 14–17, 26–29; [3] pp. 52–60	4
3	Types of relationships. Modeling simple dependencies and single inheritance. Association extensions. Modeling dependency relationships. Modeling single inheritance. Multiple inheritance. Modeling structural relationships. Creating relationship networks. Types of extended relationships. Modeling relationships of different levels. Stereotypes of generalization relationships between classes. Visibility of classes and objects. Class associations. The concept of an interface. Types of interfaces. Modeling static and dynamic types. Ref.: [4] pp. 13–24; [2] pp. 16–29; [6] pp. 18–28	4
Topic 3. Basic data structures		

Lecture No.	List of Lecture Topics and Annotations	Hours
4	Use case modeling diagram elements. Modeling system behavior. Implementing use cases using cooperation. Use case diagrams. Modeling system context. Modeling system requirements. Forward and reverse engineering. Ref.: [2] pp. 33–37; [4] pp. 28–47	4
5	Description of a usage diagram. Components of a usage diagram. Subject. Use case diagrams in design tasks. Modeling the system context. Modeling system requirements. Forward and reverse engineering. Use cases and testing Ref.: [1] pp. 30–40; [3] pp. 6–9; [4] pp. 18–28	4
6	Elements of object interaction. Prototypical instances of classes. Control flows. Behavior phases. Interaction diagrams when modeling control flows. Interaction context. Objects and roles in interaction. Prototypical objects and prototypical relationships or links. Message types and instances Ref.: [5] pp. 13–24; [2] pp. 16–19; [3] pp. 18–28	4
Topic 4. Hash tables		
7	Signal, call, time and change events. Event types. Signal series modeling. Signal class modeling. Call event synchronicity. Time event. Change event. Exception modeling. Event processing in active and passive objects. States, transitions, activities. Object life cycle modeling. Creating well-structured algorithms.. Ref.: [1] pp. 83–85; [2] pp. 50–59; [3] pp. 47–62	4
Topic 5. Complex data structures		
8	System architecture. Architecture. Architectural views. Software development life cycle. Requirements analysis and planning. Implementation. Ref.: [1] pp. 85–112; [3] pp. 148–251	4
Total for the semester		32

5.2. CONTENT OF LABORATORY WORKS

Topic No.	Laboratory Session Topic	Hours
1	Development of functional and object model domain Ref.: [1] pp. 30–40; [2] pp. 6–9; [3] pp. 18–28	6
2	Use case development. Ref.: [4] pp. 25–35	6

Topic No.	Laboratory Session Topic	Hours
3	Developing a class diagram Ref.: [4] pp. 65–78	4
4	Developing an object diagram Ref.: [1] pp. 123–144; [4] pp. 93–99	4
5	Building an interaction diagram Ref.: [1] pp. 166–184	4
6	Building an activity diagram. Ref.: [2] pp. 132–138	4
7	Building a state diagram. Ref.: [1] pp. 453–474	4
8	Development of design patterns Ref.: [1] pp. 476–507	6
Total for the semester		34

5.3. CONTENT OF INDEPENDENT WORK

The volume of independent work in the discipline " Software Modelling and Evaluation " is 84 hours. They include the study of lecture material, theoretical and laboratory tasks, preparation for laboratory work, their defense, and current testing.

Week No.	Type of Independent Work	Hours
1	Study of theoretical material from T1, preparation for Laboratory Work No. 1 Ref.: [1] pp. 30–40; [2] pp. 6–9	6
2	Study of theoretical material from T1, preparation for Laboratory Work No. 1 Ref.: [1] pp. 13–24; [2] pp. 36–39	6
3	Study of theoretical material from T2, preparation for Laboratory Work No. 2 Ref.: [1] pp. 50–60; [3] pp. 38–48	6
4	Study of theoretical material from T2, preparation for Laboratory Work No. 2 Ref.: [2] pp. 17–29; [3] pp. 81–90	6
5	Study of theoretical material from T2, preparation for Laboratory Work No. 3 Ref.: [2] pp. 61–69; [3] pp. 28–48	6
6	Study of theoretical material from T3, preparation for Laboratory Work No. 3 Ref.: [4] pp. 20–36; [2] pp. 40–54;	6
7	Study of theoretical material from T3, preparation for Laboratory Work No. 4 Ref.: [4] pp. 30–40; [5] pp. 6–9; [3] pp. 18–28	6
8	Study of theoretical material from T3, preparation for Laboratory Work No. 4. Preparation for TC No. 1 Ref.: [1] pp. 30–80; [2] pp. 16–29; [3] pp. 48–68	6

Week No.	Type of Independent Work	Hours
9	Study of theoretical material from T3, preparation for Laboratory Work No. 5 Ref.: [4] pp. 23–49; [2] pp. 6–39; [3]	6
10	Study of theoretical material from T4, preparation for Laboratory Work No. 5 Ref.: [5] pp. 27–34; [6] pp. 76–89	6
11	Study of theoretical material from T4, preparation for Laboratory Work No. 6 Ref.: [5] pp. 30–40; [7] pp. 6–9	6
12	Study of theoretical material from T4, preparation for Laboratory Work No. 6 Ref.: [5] pp. 34–65; [2] pp. 46–59	6
13	Study of theoretical material from T4, preparation for Laboratory Work No. 7 Ref.: [6] pp. 17–24; [7] pp. 8–19	6
14	Study of theoretical material from T5, preparation for Laboratory Work No. 7 Ref.: [1] pp. 74–84; [2] pp. 76–93	6
15	Study of theoretical material from T5, preparation for Laboratory Work No. 8 Ref.: [8] pp. 30–40	6
16	Study of theoretical material from T5, preparation for Laboratory Work No. 8. Preparation for TC No. 2 Ref.: [8] pp. 39–47	6
17	Study of theoretical material from T5, preparation for the final exam Ref.: [8] pp. 80–94	4
Total:		**84**

Notes: TC – Test Control, T1–T7 – Topics of the course.

6. TECHNOLOGIES AND TEACHING METHODS

The learning process for the course is based on the use of both traditional and modern teaching technologies and methods, in particular: lectures (using visualisation methods, problem-based and interactive learning, motivational techniques, and information and communication technologies); laboratory works (using training exercises, problem situation analysis, explanation, discussions, etc.); independent work (study of theoretical material, preparation for laboratory works, ongoing and final assessment), with the use of information and computer technologies and distance learning technologies.

7. METHODS OF ASSESSMENT

Ongoing assessment is carried out during practical classes, as well as on the days of control activities established by the working programme and the academic schedule.
The following methods of ongoing assessment are used:
– test-based assessment of theoretical material;
– evaluation of the results of laboratory work defence.
When determining the final semester grade, the results of both ongoing assessment and final assessment are taken into account. The final assessment is conducted on all the material of the course according to examination papers prepared in advance and approved at the meeting of the department. A student who has scored less than 60 percent of the maximum score for any type of academic work is not allowed to undergo the semester assessment until the amount of work stipulated by the Working Programme is completed. A student who has achieved a positive weighted average score (60 percent

or more of the maximum score) for all types of ongoing assessment but has failed the examination is considered to have an academic debt.
Elimination of academic debt for the semester assessment is carried out during the examination session or according to the schedule set by the dean's office in accordance with the *Regulation on Control and Assessment of Learning Outcomes of Students at Khmelnytskyi National University*.

8. COURSE POLICY

The policy of the academic course is generally determined by the system of requirements for the student as stipulated by the current University regulations on the organisation and teaching and learning support of the educational process. In particular, this includes completing the safety briefing; attendance at course classes is compulsory. For valid reasons (documentarily confirmed), theoretical training may, with the lecturer's approval, take place online. Successful completion of the course and the formation of professional competences and programme learning outcomes require preparation for each laboratory work (studying the theoretical material for the topic of the work), active participation during the class, thorough preparation of the report, defence of the results, participation in discussions regarding the constructive decisions made during the laboratory works, etc.
Students must meet the established deadlines for completing all types of academic work in accordance with the Working Programme of the course. A missed laboratory class must be completed within the deadline set by the lecturer, but no later than two weeks before the end of the theoretical classes in the semester.
The student's mastery of the theoretical material of the course is assessed through testing.
When performing laboratory work, the student must comply with the policy of academic integrity (cheating, plagiarism — including with the use of mobile devices — is prohibited). If a violation of academic integrity is detected in any type of academic work, the student receives an unsatisfactory grade and must re-do the task on the relevant topic (type of work) as stipulated by the Working Programme. Any form of academic dishonesty is unacceptable.
Within the framework of studying the course, students are provided with recognition and crediting of learning outcomes acquired through non-formal education, available on accessible platforms (https://prometheus.org.ua/, https://www.coursera.org/), which contribute to the formation of competences and the deepening of learning outcomes defined in the Working Programme of the course, or ensure the study of a relevant topic and/or type of work from the course syllabus (for more details, see the *Regulation on the Procedure for Recognition and Crediting of Learning Outcomes of Students at Khmelnytskyi National University*).

9. ASSESSMENT OF STUDENTS' LEARNING OUTCOMES DURING THE SEMESTER

Assessment of a student's academic achievements is carried out in accordance with the *Regulation on the Control and Assessment of Students' Learning Outcomes at Khmelnytskyi National University*.
During the ongoing assessment of the work performed by the student for each structural unit and the results obtained, the lecturer awards a certain number of points as set out in the Working Programme for that type of work.
Each structural unit of academic work may be credited only if the student has scored at least 60 percent (the minimum level for a positive grade) of the maximum possible points assigned to that structural unit.
When assessing students' learning outcomes for any type of academic work (structural unit), it is recommended to use the generalised criteria provided below:

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 fragmented, unstructured knowledge, cannot distinguish between main and secondary ideas, makes conceptual errors, misinterprets definitions, presents material in a chaotic and unconfident manner, and cannot apply knowledge to solve practical problems. An unsatisfactory grade is typically given to a student who is unable to continue learning the subject without additional study.

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

<table>
<tr><td colspan="8"><u>In-Class Work</u></td><td><u>Assessment Activities</u></td><td colspan="2"><u>Semester Final Assessment</u></td></tr>
<tr><td colspan="8"><u>Laboratory Work</u> №:</td><td>Test control:</td><td rowspan="2">Exam</td><td rowspan="2">Total</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>T 1-6</td></tr>
<tr><td colspan="9"><u>Number of points per type of academic work (min–max)</u></td><td rowspan="2">60-100*</td></tr>
<tr><td>3-5</td><td>3-5</td><td>3-5</td><td>3-5</td><td>3-5</td><td>3-5</td><td>3-5</td><td>3-5</td><td>6-10</td><td>24-40</td></tr>
<tr><td colspan="8">24-40</td><td>12-20</td><td>24-40</td></tr>
</table>

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**".

Assessment of Laboratory Work Defence Results

A laboratory work completed and formatted in accordance with the requirements established in the Methodological Guidelines is comprehensively assessed by the lecturer during its defence based on the following criteria:

– independence and accuracy of execution;

– completeness of the answer and understanding of the principles of building machine learning models;

– ability to justify the choice of algorithm or method;

– correctness of model implementation in the Python programming environment using appropriate libraries;

– ability to interpret the results of modelling and evaluate their suitability for solving the given task.

When assessing a laboratory session, the lecturer uses the generalised criteria outlined in the table "Assessment Criteria for Student Learning Outcomes" (minimum passing score – 3 points, maximum – 5 points).

If the student demonstrates a knowledge level below 60 percent of the maximum score established in the Working Programme for each structural unit, the laboratory work is not credited. In such a case, the student must study the topic more thoroughly, review the methodology, correct major mistakes, and re-defend the work at the time set by the lecturer.

Assessment of Test-Based Control Results

Each test included in the Working Programme consists of 30 test items, each carrying equal weight. According to the table for structuring types of academic work, the student may receive between 3 and 5 points depending on the number of correct answers.

Distribution of points depending on the number of correct answers to test items:

The test duration is 30 minutes. Students complete the test online in the Modular Learning Environment.

If a failing grade is received, the test must be retaken before the next scheduled assessment.

Distribution of points depending on correct answers to test questions

Number of Correct Answers	1-17	18-23	24-26	27-30
Percentage of Correct Answers	0-59	60-79	80-89	90-100
Number of Points	-	3	4	5

The final semester grade according to the institutional grading scale and the ECTS grading scale is determined automatically after the lecturer enters the assessment results in points for all types of

academic work into the electronic gradebook. The correspondence between the institutional grading scale and the ECTS grading scale is provided in the table "Correspondence" below.

Assessment of the Final Semester Control (Exam)

The educational programme provides for a final semester control in the form of an examination, the purpose of which is to systematically and objectively assess both the theoretical and practical preparation of the student in the course. The examination is conducted according to examination papers prepared in advance and approved at the meeting of the department. In accordance with this, the examination paper contains a combination of both theoretical questions (including in test form) and practical tasks.

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

Type of Task	**For each individual type of task**		
	Minimum (Satisfactory) Score	***Potential Positive Score (Good)****	**Maximum (Excellent) Score**
Theoretical Question № 1	3	4	5
Theoretical Question № 2	3	4	5
Practical Tasks *(6 tasks worth 3 points each)*	18	24	30
Total:	**24**	**32**	**40**

Note. ***A passing score for the exam, different from the minimum (24 points) and the maximum (40 points), falls within the range of 25–39 points and is calculated as the sum of points for all structural elements (tasks) of the exam.***

For each individual type of task in the final semester assessment, the assessment criteria for student learning outcomes provided above (see **Table – Assessment Criteria for Student Learning Outcomes**) are applied.

The final semester grade according to the institutional grading scale and the ECTS grading scale is determined automatically after the lecturer enters the assessment results in points for all types of academic work into the electronic gradebook. The correspondence between the institutional grading scale and the ECTS grading scale is shown below in the **Correspondence Table**.

The final examination grade is recorded if the total number of points accumulated by the student in the course as a result of ongoing assessment falls within the range of 60 to 100 points. In this case, a grade of *Excellent/Good/Satisfactory* is assigned according to the institutional scale, and a letter grade is assigned according to the ECTS scale, corresponding to the total number of points earned by the student as specified in the **Correspondence Table**.

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

<table>
<tr><th rowspan="2">ECTS Grade</th><th rowspan="2">Rating Scale (Points)</th><th colspan="2">Institutional Grade(Level of Achievement of the Intended Learning Outcomes in the Course)</th></tr>
<tr><th>Pass/ Fail</th><th>Exam / Graded Credit</th></tr>
<tr><td>A</td><td>90-100</td><td rowspan="4">Pass</td><td>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.</td></tr>
<tr><td>B</td><td>83-89</td><td rowspan="2">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.</td></tr>
<tr><td>C</td><td>73-82</td></tr>
</table>

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. What is software evaluation?
2. What is software modeling?
3. Why is software evaluation important in the software development life cycle?
4. What are the main objectives of software modeling?
5. What are the key differences between software modeling and software design?
11. What is UML and what is its purpose?
12. Name five types of UML diagrams.
13. What is the difference between a class diagram and an object diagram?
14. What is a use case diagram used for?
15. How is a sequence diagram different from an activity diagram?
16. What is static analysis in software evaluation?
17. What is dynamic analysis in software evaluation?
18. What is the purpose of code reviews?
19. What is formal verification?
20. Describe the difference between black-box and white-box testing.
21. What is function point analysis?
22. What is cyclomatic complexity and how is it calculated?
23. What is the role of software measurement in evaluation?
24. Name some tools used for software metrics collection.
25. How can software metrics improve software quality?
26. What is a software process model? Give examples.
27. What is a domain model?
28. What is the difference between logical and physical models?
29. Explain the Model-View-Controller (MVC) architecture.
30. What is the purpose of data flow diagrams (DFDs)?
31. How is software evaluation handled in Agile development?
32. What are some challenges in evaluating software in a DevOps environment?
33. How can continuous integration help in software evaluation?
34. What is test-driven development (TDD) and how does it relate to evaluation?
35. What is model-driven engineering (MDE)?
36. What is model transformation?

37. What is the role of simulation in software modeling?
38. What is the difference between descriptive and prescriptive models?
39. How can machine learning be used in software evaluation?
40. Describe a real-world scenario where software modeling improved the software development process.

11. EDUCATIONAL AND METHODOLOGICAL SUPPORT

The educational process for the course *"*Software Modelling and Evaluation*"* is supported with all necessary instructional and methodological materials, which are available in the Modular Learning Environment MOODLE:

1. Course *"*Software Modelling and Evaluation*"*: https://msn.khmnu.edu.ua/course/view.php?id=4201
2. Methodological Guidelines for Laboratory Sessions: https://msn.khmnu.edu.ua/course/view.php?id=4201

12. RECOMMENDED LITERATURE

Primary

1. Sommerville, I. Software Engineering (10th ed.). Pearson Education, 2021. – 816 p.
2. Pohl, K., Rupp, C. Requirements Engineering: Fundamentals, Principles, and Techniques. Springer, 2022. – 400 p.
3. Tavana, M., Pick, K. W. Managing Requirements Knowledge. Springer, 2022. – 300 p.
4. Laplante, P., Kassab, M. Requirements Engineering for Software and Systems (4th ed.). CRC Press, 2022. – 420 p.
5. La Rocca, M. Advanced Algorithms and Data Structures. Manning Publications, 2021. – 978 p.
6. D'Andrade, Brian. Software Engineering: Artificial Intelligence, Compliance, and Security. Nova Science Publishers, 2021. – 512 p.
7. Mejia, J., Muñoz, M., Rocha, A., Quiñonez, Y. (Eds.). New Perspectives in Software Engineering: Proceedings of the 9th International Conference on Software Process Improvement (CIMPS 2020). Springer, 2021. – 350 p.
8. О. О. Петрова, Г. В. Солодовник. Алгоритмічні задачі та їх вирішення. Харків: ХНУМГ ім. О. М. Бекетова, 2021. – 105 с.

Supplementary

1. ISO/IEC/IEEE 25010:2023 – Systems and Software Quality Requirements and Evaluation (SQuaRE): System and Software Quality Models. IEEE Standards Association.
2. ISO/IEC/IEEE 29148:2022 – Systems and Software Engineering — Life Cycle Processes — Requirements Engineering. IEEE Standards Association.
3. Piotr Sliż. Organizacja procesowo-projektowa. Istota, modelowanie, pomiar dojrzałości. Difin, 2021. – 292 p.
4. Островський, Д. . Лисий, А., Свистун, С., Онишко, О., Сергеєв, Є. (2023). Метод, моделювання та оцінювання прискорення об'єднання масивів з інтегрованим індексом значень *Measuring and computing devices in technological processes*, (3).– 2023. – № 4. – С. 93-99.

13. INFORMATION RESOURCES

1. Electronic Library of the University. [Electronic resource]. – Access: http://library.khmnu.edu.ua/
2. Institutional Repository of Khmelnytskyi National University. [Electronic resource]. – Access: http://elar.khmnu.edu.ua/jspui/?locale=uk
3. Modular Learning Environment. [Electronic resource]. – Access: https://msn.khmnu.edu.ua/