

ОРГАНІЗАЦІЯ НАВЧАННЯ

Тема, опис теми	Вид заняття	Оцінювання за тему	Форми і методи навчання/питання до самостійної роботи
Тема 1 Інтелектуалізація процесів інженерії програмного забезпечення як науковий напрям <u>Рекомендовані джерела:</u> 1-2			
Заняття 1.1 Інтелектуалізація процесів розробки програмного забезпечення, еволюція підходів та наукові парадигми.	Лекція 1 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 1.2 Виявлення та формулювання дослідницьких проблем, пов'язаних з інтелектуалізацією процесів інженерії програмного забезпечення у межах дисертаційної теми.	Практичне заняття 1 4 години	4 бали	Усне опитування, навчально-наукова дискусія, презентація результатів дослідження.
Заняття 1.3 Генеративні підходи як об'єкт наукового аналізу. Потенціал, обмеження, валідність результатів.	Лекція 2 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 1.4 Критичний аналіз сучасних досліджень щодо використання генеративних підходів у інженерії програмного забезпечення.	Практичне заняття 2 4 години	5 балів	Усне опитування по матеріалу лекції, навчально-наукова дискусія, робота в командах з використанням методів фасилітації, робота з науковими джерелами, презентація результатів дослідження.
Тема 2. Методології емпіричних і систематичних досліджень в інженерії програмного забезпечення <u>Рекомендовані джерела:</u> 3-5			
Заняття 2.1 Evidence-based software engineering та систематичні огляди як основа наукових досліджень.	Лекція 3 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 2.2 Формування research questions та побудова PRISMA-стратегії для дисертаційного дослідження.	Практичне заняття 3 4 години	4 бали	Навчально-наукова дискусія, презентація результатів дослідження.
Заняття 2.3 Емпіричні методи досліджень — експерименти, кейс-стаді, реплікація.	Лекція 4 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 2.4. Ідентифікація загроз валідності та методи їх мінімізації у власному дослідженні.	Практичне заняття 4 4 години	5 балів	Усне опитування, навчально-наукова дискусія, ситуаційне завдання, презентація результатів дослідження.

Тема 3. Якість програмного забезпечення та еволюція програмних систем**Рекомендовані джерела:** 6-8

Заняття 3.1 Якість програмного забезпечення як наукове поняття, моделі, метрики та обмеження вимірювання.	Лекція 5 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 3.2 Аналіз підходів до оцінювання якості програмного забезпечення в існуючих наукових дослідженнях.	Практичне заняття 5 4 години	4 бали	Усне опитування, робота з науковими джерелами, навчально-наукова дискусія, робота в командах, презентація результатів дослідження.
Заняття 3.3 Технічний борг і еволюція програмних систем як предмета наукового аналізу.	Лекція 6 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 3.4 Формування гіпотез щодо впливу еволюційних змін на якість програмного забезпечення у межах дисертаційного дослідження.	Практичне заняття 6 4 години	5 балів	Усне опитування по матеріалу лекції, навчально-наукова дискусія, робота в командах, презентація результатів дослідження.

Тема 4. Архітектурні рішення та керування життєвим циклом програмного забезпечення**Рекомендовані джерела:** 9-10

Заняття 4.1 Архітектура програмного забезпечення як наукова гіпотеза. Компроміси, припущення, довгострокові наслідки.	Лекція 7 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 4.2 Аналіз архітектурних рішень у сучасних дослідженнях інженерії програмного забезпечення.	Практичне заняття 7 2 години	4 бали	Усне опитування, робота з науковими джерелами, навчально-наукова дискусія, презентація результатів дослідження.
Заняття 4.3 Керування життєвим циклом програмного забезпечення. Наукові підходи, моделі та автоматизація як дослідницький аспект.	Лекція 8 2 години		Усне опитування, навчально-наукова дискусія, презентація результатів дослідження.
Заняття 4.4 Формування індивідуального дослідницького напрямку, пов'язаного з архітектурою або життєвим циклом програмного забезпечення.	Практичне заняття 8 2 години	5 балів	Усне опитування, навчально-наукова дискусія, презентація результатів дослідження.
Заняття 4.5 Сучасні виклики архітектури програмного забезпечення, перехідні аспекти, сталість, розвиток.	Лекція 9 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 4.6 Презентація та наукове обговорення результатів власного дослідження з архітектурних аспектів або життєвого циклу програмного забезпечення.	Практичне заняття 9 2 години	6 балів	Усне опитування, навчально-наукова дискусія, презентація результатів дослідження.

Самостійна робота			
1. Аналіз сучасних наукових напрямів в інженерії програмного забезпечення	9 годин	3 бали	Виокремлення ключових дослідницьких підходів, проблемних зон та відкритих наукових питань у межах власної теми дисертаційного дослідження.
2. Опрацювання методологій емпіричних досліджень	9 годин	3 бали	Методології емпіричних досліджень, систематичних оглядів і систематичних мапінг-досліджень у галузі інженерії програмного забезпечення. Аналіз прикладів застосування PRISMA-підходів та загроз валідності у сучасних наукових роботах.
3. Формування дослідницьких питань та гіпотез	9 годин	3 бали	Розробка та уточнення дослідницьких питань і гіпотез у контексті власного дисертаційного дослідження. Узгодження сформульованих положень з існуючими науковими підходами та виявлення потенційних дослідницьких лакун.
4. Аналіз літератури з проблем якості та еволюції програмного забезпечення	9 годин	3 бали	Опрацювання наукових публікацій, присвячених моделям якості програмного забезпечення, метрикам та технічному боргу. Порівняльний аналіз підходів до оцінювання якості та інтерпретації результатів досліджень.
5. Дослідження архітектурних аспектів та життєвого циклу програмного забезпечення	9 годин	3 бали	Аналіз сучасних наукових праць, що розглядають архітектурні рішення та моделі керування життєвим циклом програмного забезпечення. Виявлення архітектурних компромісів та їх впливу на еволюцію програмних систем.
6. Прикладні аспекти використання перспективних технологій інженерії програмного забезпечення.	9 годин	3 бали	Підготовка до виконання індивідуальних завдань на практичних заняттях. Підбір прикладів практичного застосування методів та інструментів за темою. Підготовка короткої доповіді за темою для обговорення в дискусійному форматі. Критична оцінка обраної методології дослідження та формування бачення подальших напрямів розвитку дисертаційної роботи.
МАТЕРІАЛЬНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ			
• Мультимедійний проектор. • Комп'ютерний клас для проведення практичних занять. • Програмне забезпечення: Microsoft Office 365, інше програмне забезпечення визначається характером досліджень здобувачів в рамках виконання індивідуальних завдань.			
ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ			
Основна література: 1. Machine Learning Pipeline for Software Engineering: A Systematic Literature Review, Samah Kansab (2025). <i>arXiv</i>. https://arxiv.org/abs/2508.00045. 2. Software Architecture Meets LLMs: A Systematic Literature Review, Schmid, L., Hey, T., Koziolk, A., et al. (2025). <i>arXiv</i>. https://arxiv.org/abs/2505.16697 3. Evidence-based Software Engineering Guidelines Revisited Pfleeger, S. L., & Kitchenham, B. (2025). <i>IEEE Transactions on Software Engineering</i>. https://doi.org/10.1109/TSE.2025.3526730			

4. Empirical research in software architecture — Perceptions of the community, Galster, M., Weyns, D., et al. (2023). *Journal of Systems and Software*. <https://doi.org/10.1016/j.jss.2023.111684>
5. An empirical study on self-admitted technical debt in modern code review. *Information and Software Technology* (2022). <https://doi.org/10.1016/j.infsof.2022.106855>
6. Software design analysis and technical debt management based on design rule theory. *Information and Software Technology* (2023). <https://doi.org/10.1016/j.infsof.2023.107322>
7. Systematic literature review on forecasting and prediction of technical debt evolution, Ajibode, A., Apedo, Y., Ajibode, T. (2024). *arXiv*. <https://arxiv.org/abs/2406.12026>
8. Tracing the Lifecycle of Architecture Technical Debt in Software Systems: A Dependency Approach, Sutoyo, E., Avgeriou, P., Capiluppi, A. (2025). *arXiv*. <https://arxiv.org/abs/2501.15387>
9. Artificial Intelligence for Software Architecture: Literature Review and the Road Ahead, Bucaioni, A., Weyssow, M., He, J., Lyu, Y., Lo, D. (2025). *arXiv*. <https://arxiv.org/abs/2504.04334>
10. Empirical research in software architecture — Perceptions of the community, Galster, M., Weyns, D., et al. (2023). *Journal of Systems and Software*. <https://doi.org/10.1016/j.jss.2023.111684>
11. Вибрані питання інженерії програмного забезпечення: навч. посіб. / МОН України, Уманський держ. пед. ун-т імені Павла Тичини ; уклад Г. В. Ткачук. — Умань : ВПЦ «Віззаві», 2021. — 191 с.
12. Romina Eramo, Michele Tucci, Daniele Di Pompeo, Vittorio Cortellessa, Antiniscia Di Marco, Davide Taibi, Architectural support for software performance in continuous software engineering: A systematic mapping study, *Journal of Systems and Software*, Volume 207, 2024, 111833, ISSN 0164-1212, <https://doi.org/10.1016/j.jss.2023.111833>.
13. Cathy Guevara-Vega, Beatriz Bernárdez, Margarita Cruz, Amador Durán, Antonio Ruiz-Cortés, Martin Solari, Research artifacts for human-oriented experiments in software engineering: An ACM badges-driven structure proposal, *Journal of Systems and Software*, Volume 218, 2024, 112187, ISSN 0164-1212, <https://doi.org/10.1016/j.jss.2024.112187>.
14. Achim D. Brucker, Idir Ait-Sadoune, Nicolas Méric, Burkhardt Wolff, Parametric ontologies in formal software engineering, *Science of Computer Programming*, Volume 241, 2025, 103231, ISSN 0167-6423, <https://doi.org/10.1016/j.scico.2024.103231>.
15. Manal Kharbouch, Aurora Vizcaino, José Alberto García-Berná, Félix García, Ambrosio Toval, Oscar Pedreira, Ali Idri, José Luis Fernández-Alemán, Uncharted dimensions, gaps, and future trends of serious games in software engineering, *Computer Standards & Interfaces*, Volume 92, 2025, 103915, ISSN 0920-5489, <https://doi.org/10.1016/j.csi.2024.103915>.
16. Boubou Thiam Niang, Giacomo Kahn, Yacine Ouzrout, Mustapha Derras, Jannik Laval, PhaDOP: A Pharo framework for implementing software product lines using Delta-Oriented Programming and model-based engineering, *Journal of Computer Languages*, Volume 80, 2024, 101283, ISSN 2590-1184, <https://doi.org/10.1016/j.cola.2024.101283>.
17. Kateryna Kolesnikova, Dmytro Lukianov, Askar Khikmetov, Kaisar Alpysbayev, Ardak Mukhamedyeva, Yergali Dauletbek, Using Model-Based Systems Engineering Tools in Software Development, *Procedia Computer Science*, Volume 241, 2024, Pages 470-475, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2024.08.066>.
18. Morteza Noferesti, Naser Ezzati-Jivan, Enhancing empirical software performance engineering research with kernel-level events: A comprehensive system tracing approach, *Journal of Systems and Software*, Volume 216, 2024, 112117, ISSN 0164-1212, <https://doi.org/10.1016/j.jss.2024.112117>.
19. Jürgen Börstler, Nauman bin Ali, Kai Petersen, Emelie Engström, Acceptance behavior theories and models in software engineering — A mapping study, *Information and Software Technology*, Volume 172, 2024, 107469, ISSN 0950-5849, <https://doi.org/10.1016/j.infsof.2024.107469>.
20. Thilagavathy R, Veeramani T, Sundaravadivazhagan B, Deebalakshmi R, Evolution of software engineering: From traditional to modern approaches, *Advances in Computers*, Elsevier, 2025, ISSN 0065-2458, <https://doi.org/10.1016/bs.adcom.2025.06.017>.
21. Jaswanth Revuri, Rakesh Kumar Sakthivel, Gayathri Nagasubramanian, Ethical AI and responsible automation in software engineering, *Advances in Computers*, Elsevier, 2025, ISSN 0065-2458, <https://doi.org/10.1016/bs.adcom.2025.07.004>.
22. Aws A. Magabaleh, Lana L. Ghraibeh, Afnan Y. Audeh, A.S. Albahri, Muhammet Deveci, Jurgita Antucheviciene, Systematic review of software engineering uses of multi-criteria decision-making methods: Trends, bibliographic analysis, challenges, recommendations, and future directions, *Applied Soft Computing*, Volume 163, 2024, 111859, ISSN 1568-4946, <https://doi.org/10.1016/j.asoc.2024.111859>.

23. Yves R. Kirschner, Moritz Gstür, Timur Sağlam, Sebastian Weber, Anne Kozirolek, Retriever: A view-based approach to reverse engineering software architecture models, *Journal of Systems and Software*, Volume 220, 2025, 112277, ISSN 0164-1212, <https://doi.org/10.1016/j.jss.2024.112277>.

Додаткова література:

24. Yang Liu, Jiu hao Wang, Yahui Li, Weichuan Dong, Lanxue Dang, Research on function-behavior-structure-based software engineering methodology for software defined systems, *Space Habitation*, Volume 1, Issue 1, 2025, 100003, ISSN 2950-6166, <https://doi.org/10.1016/j.spaceh.2025.100003>.
25. Skladannyi, P., Nehodenko, O., Shevchenko, S., Zolotukhina, O., & Nehodenko, V. (2022). Modified Delta Maintainability Model of Object-Oriented Software. *Cybersecurity Providing in Information and Telecommunication Systems 2022*, 3288(1), 117-124. Scopus.
26. Лихвар А.В., Негоденко О.В., Золотухіна О.А., Шевченко С.М., Олімпієва Ю.І. Метод підвищення ефективності оцінки супроводженості об'єктно-орієнтованого програмного забезпечення. Телекомунікаційні та інформаційні технології. 2021. № 4 (73). С.4-11. <https://doi.org/10.31673/2412-4338.2021.040411>
27. Худік Б.О., Шоробура В.В. Розробка методу оптимізації програмного коду на основі генетичного алгоритму. «Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2025, №27. С. 141-149. <https://doi.org/10.28925/2663-4023.2025.27.718>
28. Асеева А., Кулаковська І. Аналіз проблем вибору технології для розробки програмного забезпечення. *Computer-integrated technologies education science production*. 2019. DOI:10.36910/6775-2524-0560-2019-37-2
29. Korzachenko, O. V. LOW-CODE та NO-CODE BPMS: Suchasni trendy avtomatyzatsii biznes-protsesiv pidpriemstva [LOW-CODE and NO-CODE BPMS: Modern trends in business process automation]. *Modeliuvannia ta Informatsiini Systemy v Ekonomitsi. Natsionalnyi Ekonomichnyi Universytet imeni Vadyma Hetmana*, 2022 (102), 102-114.
30. Кордунова , Ю., Смотров , О., Кокотко , І., & Малець , Р. (2021). Аналіз традиційного та гнучкого підходів до створення програмного забезпечення в динамічних умовах. *Управління розвитком складних систем*, (47), 71–77. <https://doi.org/10.32347/2412-9933.2021.47.71-77>
31. Коваль Г.І. Основні задачі системи підтримки прийняття рішень в інженерії надійності програмних систем. *Проблеми програмування*. 2005. № 3, 35-41.
32. Pierce B.C., Amorim A.A., Casinghino, C. Gaboardi M., Greenberg M., Hrițcu C., Sjöberg V., Yorgey B. *Software foundations*. Vol. 2 (2019) <https://softwarefoundations.cis.upenn.edu/plfcurrent/index.html>
33. Замрій, І., Шахматов, І. Підвищення безпеки вебзастосунків з допомогою інноваційних патернів інтеграції штучного інтелекту. *Сучасний стан наукових досліджень та технологій в промисловості*, 2024 , 1(27), с. 67–80. doi: 10.30837/ITSSI.2024.27.067.
34. Макарець М. О., Тушич А. М., Замрій І. В., Алексіна Л. Т. Проблеми, труднощі та можливості IoT та хмарних обчислень. *Зв'язок*, 2021, №4. – С. 20-25. <https://doi.org/10.31673/2412-9070.2021.042025>
35. Peters K. Comparing Process Calculi Using Encodings In: J.A. Pe´rez and J. Rot (Eds.): *Combined Workshop on Expressiveness in Concurrency and Structural Operational Semantics (EXPRESS/SOS 2019)*. EPTCS 300, 2019, pp. 19–38.
36. O'Regan, G. (2023). *Mathematical Foundations of Software Engineering : A Practical Guide to Essentials*. Texts in Computer Science. Springer, Cham.
37. Булига, К., Булига, О., & Коцюбівська, К. (2021). Використання методів програмної інженерії під час моделювання бізнес-проектів у сфері послуг. *Цифрова платформа: інформаційні технології в соціокультурній сфері*, 4(2), 219–227. <https://doi.org/10.31866/2617-796X.4.2.2021.247486>
38. Кулаков О.В., Замрій І.В. Інтеграція метрик відстежуваності та поведінкових метрик для оптимізації рефакторингових рішень у програмних системах. *Всеукраїнська науково-технічна конференція «Виклики та рішення в програмній інженерії»*, 26 листопада 2024 року, ДУІКТ. Збірник тез. К.: ДУІКТ, 2024. С. 71-77.
39. Притаманний В.В., Замрій І.В. Аналіз та методи покращення бібліотеки Ardalis.Specification для оптимізації архітектури .NET-додатків. *Збірник тез Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях»*, 24 квітня 2025 р. К.: ДУІКТ, 2025,С. 19-21.
40. Макаров А.Д., Замрій І.В. Побудова гнучкої архітектури через ізоляцію доменної логіки веб-застосунків. *Збірник тез Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях»*, 24 квітня 2025 р. К.: ДУІКТ, 2025,С. 69-71.

ПОЛІТИКА КУРСУ («ПРАВИЛА ГРИ»)

Відвідування

Студенти зобов'язані відвідувати заняття згідно з розкладом та вчасно інформувати викладача про неможливість із поважних причин відвідувати заняття, складати (перескладати) іспит.

Відпрацювання

Якщо студент із поважних причин був відсутній на практичному занятті, він має право його відпрацювати. Відпрацювання полягає у виконанні студентом в позанавчальний час завдання за темою заняття та захисті роботи викладачеві на наступному практичному занятті (при наявності вільного часу у викладача), або під час додаткової консультації з викладачем, час якої узгоджується окремо. Для виконання завдань курсу може використовуватись обладнання лабораторій кафедри Інженерії програмного забезпечення або власна комп'ютерна техніка студента.

Академічна доброчесність

Під час вивчення дисципліни всі учасники навчального процесу мають дотримуватися вимог, установлених Положенням про академічну доброчесність здобувачів освіти у ДУІКТ, своєю діяльністю утверджувати академічну доброчесність як засадничу цінність університетської спільноти. Студенти не повинні отримувати чи надавати недозволеної допомоги, вдаватися до жодних інших нечесних дій під час виконання навчальних завдань. Порухення студентом вимог академічної доброчесності під час опанування дисципліни тягне виставлення оцінки 0 балів (неприйнятно).

Поведінка

Спілкуючись з учасниками навчального процесу, студенти мають дотримуватися етичних норм, утримуватися від образливих проявів емоцій, бути політично та гендерно коректними й толерантними, поважати звичаї й традиції різних етнічних, культурних, соціальних груп і релігійних конфесій. Під час лекційних і практичних занять студентам забороняється приймати їжу та користуватися мобільним телефоном. Використання мобільних телефонів допускається при виконанні практичних занять для доступу до навчальних та довідкових матеріалів.

Студенти з особливими потребами

Кафедра інженерії програмного забезпечення ДУІКТ всіляко сприяє вирішенню індивідуальних проблем студентів з особливими потребами, сприймаючи їх як рівних в інтегрованому студентському колективі. Для покращення сприйняття навчального матеріалу особами з особливими потребами можуть бути використані додаткові програмні засоби, що розширюють можливості інклюзивного доступу (наприклад, програми озвучування тексту). Для осіб з порушеннями зору завдання, що передбачають використання графічних методів роботи з ризиками, можуть бути замінені на завдання з використанням табличних підходів (наприклад, представлення графових структур табличними еквівалентами).

Важливі дати

Подія (чи вид роботи)	Дата
Подання викладачам, що проводять практичні заняття, всіх відпрацьованих завдань.	не пізніше, ніж за 5 днів до призначеного іспиту
Подання лектору даних для нарахування додаткових балів (тези доповідей, статті, проекти тощо).	не пізніше, ніж за 5 днів до призначеного іспиту

КРИТЕРІЇ ТА МЕТОДИ ОЦІНЮВАННЯ

Оцінювання студентів в семестрі здійснюється за накопичувальною 100-бальною системою, складається із двох основних блоків і розподіляється в пропорції 60 (бали, напрацьовані під час вивчення дисципліни – робота на практичних заняттях та виконання завдань самостійної роботи) на 40 (іспит).

Іспит включає питання у форматі комп'ютерного тесту з автоматизованою перевіркою та відкриті питання, на які здобувач повинен надати розгорнуту та обґрунтовану відповідь. Тест містить питання, що охоплюють теоретичну та практичну складову курсу. Під час тестування заборонено використання будь-яких джерел. Бали за кожне тестове питання визначаються відповідно до складності питання і становлять від 0,2 до 1 балу. Сумарна кількість балів за екзаменаційний тест складає 10 балів. Відкриті питання оцінюються в 30 балів.

Присутність здобувача на іспиті є обов'язковою.

Студенти, які протягом семестру набрали менше 35 балів за роботу на практичних заняттях та самостійну роботу, вважаються недопущеними до іспиту.

Якщо студента не допущено до складання іспиту, йому надається час до перескладання для виконання всіх вимог допуску.

Якщо студент був допущений до складання іспиту і при цьому:

1) не набрав у підсумку необхідну кількість балів для отримання позитивної оцінки (має від 35 до 59 балів),

2) або не з'явився без поважної причини, то вважається, що він використав першу спробу скласти іспит і має заборгованість.

Студент має право на два перескладання. При повторному перескладанні екзамену його може приймати комісія, яка створюється директором ННІТ. Оцінка комісії є остаточною.

Форми контролю	Види навчальної роботи	Оцінювання	
ПОТОЧНИЙ КОНТРОЛЬ	• Виконання практичних робіт	42 бали	
	• Самостійна робота	18 балів	
ПІДСУМКОВЕ ОЦІНЮВАННЯ <i>Ispit</i>	Іспит складається з тестової форми (10 балів) та розгорнутої відповіді на відкриті питання (30 балів).	40 балів	
Додаткова оцінка			
Види навчальної роботи		Оцінювання	
Участь у наукових конференціях, підготовка наукових публікацій за тематикою освітньої компоненти:			
- тези доповіді на фаховій конференції;		2 бали	
- стаття у фаховому виданні;		5 балів	
- стаття в іноземному рецензованому виданні;		8 балів	
- Всеукраїнських та Міжнародних конкурсах наукових робіт/наукових проєктів.		до 10 балів за результатами конкурсу (враховуються лише результати, що відповідають індивідуальному завданню та самостійній роботі, вказаних в силабусі дисципліни)	
- отримання сертифікату в рамках інформальної (неформальної) освіти		до 15 балів за один курс (якщо теми курсів перетинаються між собою, то зарахування балів відбувається лише один раз; враховуються лише ті теми курсів, що корелюються з темами, вказаними в силабусі дисципліни)	
Максимальна кількість додаткових балів, які можуть бути зараховані здобувачу освіти - 10 балів.			
ПІДСУМКОВА ОЦІНКА ЗА ДИСЦИПЛІНУ			
бали	Критерії оцінювання	Рівень компетентності	Оцінка /запис в екзаменаційній відомості
90-100	Студент демонструє повні й міцні знання навчального матеріалу в обсязі, що відповідає робочій програмі дисципліни, правильно й обґрунтовано приймає необхідні рішення в різних нестандартних ситуаціях. Вміє реалізувати теоретичні положення дисципліни в практичних розрахунках, аналізувати та співставляти дані об'єктів діяльності фахівця на основі набутих з даної та суміжних дисциплін знань та умінь. Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань проявив вміння самостійно вирішувати поставлені завдання, активно включатись в дискусії, може відстоювати власну позицію в питаннях та рішеннях, що розглядаються. Зменшення 100-бальної оцінки може бути пов'язане з недостатнім розкриттям питань, що стосується дисципліни, яка вивчається, але виходить за рамки	Високий Повністю забезпечує вимоги до знань, умінь і навичок, що викладені в робочій програмі дисципліни. Власні пропозиції студента в оцінках і вирішенні практичних задач підвищує його вміння використовувати знання, які він отримав при вивченні інших дисциплін, а також знання, набуті при самостійному поглибленому вивченні питань, що відносяться до дисципліни, яка вивчається.	Відмінно / Зараховано (А)

	об'єму матеріалу, передбаченого робочою програмою, або студент проявляє невпевненість в тлумаченні теоретичних положень чи складних практичних завдань.		
82-89	Студент демонструє гарні знання, добре володіє матеріалом, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати теоретичні положення при вирішенні практичних задач, але допускає окремі неточності. Вміє самостійно виправляти допущені помилки, кількість яких є незначною. Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань та поясненні прийнятих рішень, дає вичерпні пояснення.	Достатній Забезпечує студенту самостійне вирішення основних практичних задач в умовах, коли вихідні дані в них змінюються порівняно з прикладами, що розглянуті при вивченні дисципліни	Добре / Зараховано (B)
75-81	Студент в загальному добре володіє матеріалом, знає основні положення матеріалу, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати при вирішенні типових практичних завдань, але допускає окремі неточності. Вміє пояснити основні положення виконаних завдань та дати правильні відповіді при зміні результату при заданій зміні вихідних параметрів. Помилки у відповідях/ рішеннях/ розрахунках не є системними. Знає характеристики основних положень, що мають визначальне значення при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань та поясненні прийнятих рішень, в межах дисципліни, що вивчається.	Достатній Конкретний рівень, за вивченим матеріалом робочої програми дисципліни. Додаткові питання про можливість використання теоретичних положень для практичного використання викликають утруднення.	Добре / Зараховано (C)
67-74	Студент засвоїв основний теоретичний матеріал, передбачений робочою програмою дисципліни, та розуміє постанову стандартних практичних завдань, має пропозиції щодо напрямку їх вирішень. Розуміє основні положення, що є визначальними в курсі, може вирішувати подібні завдання тим, що розглядалися з викладачем, але допускає значну кількість неточностей і грубих помилок, які може усувати за допомогою викладача.	Середній Забезпечує достатньо надійний рівень відтворення основних положень дисципліни	Задовільно / Зараховано (D)
60-66	Студент має певні знання, передбачені в робочій програмі дисципліни, володіє основними положеннями, що вивчаються на рівні, який визначається як мінімально допустимий. З використанням основних теоретичних положень, студент з труднощами пояснює правила вирішення практичних/розрахункових завдань дисципліни. Виконання практичних / індивідуальних / контрольних завдань значно формалізовано: є відповідність алгоритму, але відсутнє глибоке розуміння роботи та взаємозв'язків з іншими дисциплінами.	Середній Є мінімально допустимим у всіх складових навчальної програми з дисципліни	Задовільно / Зараховано (E)
35-59	Студент може відтворити окремі фрагменти з курсу. Незважаючи на те, що програму навчальної дисципліни студент виконав, працював він пасивно, його відповіді під час практичних робіт в більшості є невірними, необґрунтованими. Цілісність розуміння матеріалу з дисципліни у студента відсутня.	Низький Не забезпечує практичної реалізації задач, що формуються при вивченні дисципліни	Незадовільно з можливістю повторного складання) / Не зараховано (FX) <i>В залікову книжку не представляється</i>
0-34	Студент повністю не виконав вимог робочої програми навчальної дисципліни. Його знання на підсумкових етапах навчання є фрагментарними. Студент не допущений до здачі екзамену/заліку.	Незадовільний Студент не підготовлений до самостійного вирішення задач, які окреслює мета та	Незадовільно з обов'язковим повторним вивченням / Не допущений (F) <i>В залікову</i>

		завдання дисципліни	<i>книжку не проставля- ється</i>
--	--	---------------------	---------------------------------------

ПОЛІТИКА ДОБРОЧЕСНОСТІ

Здобувач вищої освіти виконуючи самостійну або індивідуальну роботу повинен дотримуватись політики доброчесності. У разі наявності плагіату в будь-яких видах робіт Здобувача він отримує незадовільну оцінку і повинен повторно виконати завдання, які передбачені у Силабусі.