

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ОБСЛУГОВУВАННЯ, РОЗВИТКУ ТА РЕІНЖІНІРИНГУ
ПРОГРАМНИХ СИСТЕМ»

Лектор курсу			Худік Богдан Олександрович, доктор філософії	Контактна інформація лектора (e-mail), курсу на Google Workspace for Education	e-mail: b.khudik@duikt.edu.ua код курсу на Google Workspace for Education: ucjntzt		
Галузь знань			F Інформаційні технології	Рівень вищої освіти	Доктор філософії		
Спеціальність			F2 Інженерія програмного забезпечення	Семестр	4		
Освітньо-наукова програма			Інженерія програмного забезпечення	Тип дисципліни	Цикл обов’язкових компонент ОНП		
Обсяг:	Кредитів ЕКТС	Годин	За видами занять				
			Лекцій		Семінарських занять	Практичних занять	Лабораторних занять
	3	90	14	-	28	-	48

АНОТАЦІЯ КУРСУ

Взаємозв'язок у структурно-логічній схемі

Освітні компоненти, які передують вивченню		1. Перспективні методи та технології програмної інженерії. 2. Методи та технології науково-інноваційної діяльності в програмній інженерії 3. Методи та засоби антропоцентричних обчислень та технологій
Освітні компоненти для яких є базовою		-
Мета курсу:	формування у здобувачів системного наукового бачення процесів супроводу, еволюції та реінжинірингу програмного забезпечення, а також здатності критично осмислювати та розвивати сучасні архітектурні, технологічні й інструментальні підходи в інженерії програмного забезпечення.	

Компетентності відповідно до освітньої програми

Soft- skills / Загальні компетентності (ЗК)	Hard-skills / Спеціальні компетентності (СК)
ЗК02. Здатність розв'язувати комплексні проблеми у сфері інженерії програмного забезпечення та з дотичних до неї міждисциплінарних напрямках на основі системного наукового світогляду та загального культурного кругозору із дотриманням принципів професійної етики та академічної доброчесності.	СК04. Здатність відстежувати тенденції розвитку інженерії програмного забезпечення та критично переосмислювати наявні технології. СК05. Здатність до розроблення нових та вдосконалення існуючих моделей, методів, засобів, процесів у сфері інженерії програмного забезпечення, які забезпечують розвиток або надають нові можливості технологіям розробки та супроводження програмного забезпечення.

Програмні результати навчання (РН)

РН03. Пропонувати нові ефективні методи і моделі розроблення, впровадження, супроводу та забезпечення якості програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.

РН05. Застосовувати сучасні інструменти і технології пошуку, оброблення та аналізу інформації, зокрема, статистичні методи аналізу даних великого обсягу та/або складної структури, спеціалізовані бази даних та інформаційні системи для покращення ефективності програмних систем.

РН06. Формулювати і перевіряти гіпотези; використовувати для обґрунтування висновків належні докази, зокрема, результати теоретичного аналізу, експериментальних досліджень і математичного та/або комп'ютерного моделювання, наявні літературні дані.

РН09. Формулювати та вирішувати задачі оптимізації, адаптації, прогнозування, керування та прийняття рішень щодо процесів, засобів та ресурсів розробки, впровадження, супроводу та експлуатації програмного забезпечення.

ОРГАНІЗАЦІЯ НАВЧАННЯ

Тема, опис теми	Вид занять	Оцінювання за тему, балів	Форми і методи навчання/питання до самостійної роботи
Тема 1. Архітектурні та інструментальні засади супроводу програмного забезпечення <u>Рекомендовані джерела: 1-4</u>			
Заняття 1.1. Архітектурні основи супроводу та еволюції програмного забезпечення	Лекція 1 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 1.2. Аналіз архітектури програмного забезпечення з позицій супроводжуваності	Практичне заняття 1 2 години	3	Усне опитування, виконання завдань на практичне застосування знань і вмінь.
Заняття 1.3. Виявлення архітектурного технічного боргу	Практичне заняття 2 2 години	3	Індивідуальні завдання, виконання завдань на практичне застосування знань і вмінь.
Заняття 1.4. Технології та інструменти обслуговування програмного забезпечення	Лекція 2 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 1.5. Оцінювання інструментів статичного аналізу коду	Практичне заняття 3 2 години	3	Усне опитування, виконання завдань на практичне застосування знань і вмінь.
Заняття 1.6. Інструментальна підтримка еволюції програмного забезпечення в CI/CD	Практичне заняття 4 2 години	3	Усне опитування по матеріалу лекції, навчально-наукова дискусія, робота з науковими джерелами, презентація результатів дослідження.
Тема 2. Реінжиніринг програмного забезпечення: моделі, процеси та архітектурні трансформації <u>Рекомендовані джерела: 5-8</u>			
Заняття 2.1. Реінжиніринг програмного забезпечення: концепції та процеси	Лекція 3 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія

Заняття 2.2. Аналіз доцільності реінжинірингу програмної системи	Практичне заняття 5 2 години	3	Усне опитування, виконання завдань на практичне застосування знань і вмінь.
Заняття 2.3. Реверс-інжиніринг та відновлення знань про систему	Практичне заняття 6 2 години	3	Усне опитування по матеріалу лекції, навчально-наукова дискусія, робота з науковими джерелами, презентація результатів дослідження.
Заняття 2.4. Архітектурний та модельно-орієнтований реінжиніринг	Лекція 4 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 2.5. Архітектурний реінжиніринг існуючої системи	Практичне заняття 7 2 години	3	Усне опитування, виконання завдань на практичне застосування знань і вмінь.
Заняття 2.6. Модельно-орієнтований реінжиніринг	Практичне заняття 8 2 години	3	Усне опитування по матеріалу лекції, навчально-наукова дискусія, робота з науковими джерелами, презентація результатів дослідження.
Тема 3. Емпіричні, аналітичні та орієнтовані на дані методи дослідження еволюції програмного забезпечення <u>Рекомендовані джерела: 1-13</u>			
Заняття 3.1. Емпіричні та орієнтовані на дані методи аналізу еволюції програмного забезпечення	Лекція 5 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 3.2. Mining Software Repositories: аналіз еволюції	Практичне заняття 9 2 години	3	Усне опитування, виконання завдань на практичне застосування знань і вмінь.
Заняття 3.3. Аналіз стандартів супроводу та еволюції програмного забезпечення	Практичне заняття 10 2 години	3	Усне опитування по матеріалу лекції, навчально-наукова дискусія, робота з науковими джерелами, презентація результатів дослідження.
Заняття 3.4. Прогнозування, оптимізація та підтримка рішень у супроводі програмного забезпечення	Лекція 6 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 3.5. Прогнозування дефектів і деградації якості	Практичне заняття 11 2 години	3	Усне опитування, виконання завдань на практичне застосування знань і вмінь.
Заняття 3.6. Оптимізація ресурсів супроводу програмного забезпечення	Практичне заняття 12 2 години	3	Усне опитування по матеріалу лекції, навчально-наукова дискусія, робота з науковими джерелами, презентація результатів дослідження.

Заняття 3.7. Управління еволюцією та стратегічний розвиток програмного забезпечення	Лекція 7 2 години		Пояснювально-ілюстративний, лекція-візуалізація, навчально-наукова дискусія
Заняття 3.8. Аналіз стратегій еволюції програмного забезпечення	Практичне заняття 13 2 години	3	Усне опитування, виконання завдань на практичне застосування знань і вмінь.
Заняття 3.9. Управління ризиками та технічним боргом	Практичне заняття 14 2 години	3	Усне опитування по матеріалу лекції, навчально-наукова дискусія, робота з науковими джерелами, презентація результатів дослідження.
	Самостійна робота		
Тема 1. Архітектурні та інструментальні засади супроводу програмного забезпечення Тема 2. Реінжиніринг програмного забезпечення: моделі, процеси та архітектурні трансформації Тема 3. Емпіричні, аналітичні та орієнтовані на дані методи дослідження еволюції програмного забезпечення	16 годин	6	1. Аналіз state-of-the-art інструментів і методів.
	16 годин	6	2. Оцінка можливості використання великих даних і спеціалізованих БД для оцінки ефективності ПЗ. 3. Оптимізація рішень щодо супроводу і розвитку ПЗ.
	16 годин	6	4. Адаптивні та прогностичні підходи до еволюції ПЗ.
МАТЕРІАЛЬНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ			
Мультимедійне обладнання. Навчальні аудиторії з доступом в мережі Інтернет. Програмне забезпечення: Git + GitHub; GitLab; SonarQube Community Edition; Archi; Modelio; IDE з підтримкою рефакторингу (Eclipse або IntelliJ IDEA Community); Python (open source); Open-source програмні проєкти (Apache, Linux, PostgreSQL).			
ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ			
Основні джерела 1. Мартін Р. С. Чистий код: створення і рефакторинг за допомогою Agile – Харків: «Ранок»: «Фабула», 2019. 2. Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D. Refactoring: Improving the Design of Existing Code. Addison-Wesley Professional, 2nd ed., 2018. 3. Bouchiha D. Reengineering Legacy Systems Towards New Technologies. Encyclopedia of Information Science and Technology, Fifth Edition, IGI Global. 2020. Pp. 1214–1230. DOI: 10.4018/978-1-7998-3479-3.ch084 4. Team N. Reverse Engineering. Handbook for CTFers, Singapore: Springer Nature Singapore. 2022. Pp. 295–427. DOI: 10.1007/978-981-19-0336-6_5 5. Business Process Model and Notation (BPMN). Web-site. URL: https://www.omg.org/spec/BPMN/2.0/PDF . 6. Jon Holt. A Pragmatic Guide to Business Process Modelling. British Computer Society; 2nd edition, 2019. 220 p. 7. Gerardus Blokdyk. IDEF Third Edition. 5STARCooks, 2022. 299 p. 8. Козир С.В., Слесарєв В.В., Ус С.А., Хом'як Т.В.. Моделювання та реінжиніринг бізнеспроцесів: підручн. М-во освіти і науки України; Нац. техн.			

ун-т «Дніпровська політехніка», Дніпро: НТУ «ДП», 2022. 163 с.

9. Gerardus Blokdyk. Structured Analysis And Design Technique A Complete Guide. STARCooks, 2021. 314 p.

10. ДСТУ ISO 9001-2001. Системи управління якістю вимоги. Web-site. URL: https://dnaop.com/html/43926/doc-%D0%94%D0%A1%D0%A2%D0%A3_ISO_9001-2001

11. Великодній С. С., Тимофєєва О. С. Реінжиніринг програмного забезпечення інформаційних систем: монографія / С. С. Великодній, О. С. Тимофєєва. – Одеса: Видавничий дім «Гельветика», 2020. – 160 с. / Режим доступу: http://eprints.library.odeku.edu.ua/id/eprint/8232/1/VelykodniySS_Reinginiiring_programn_ogo_zabezpechenia_2020.pdf

12. Рефакторинг. Поліпшення існуючого коду / М. Фаулер, К. Бек, Дж. Брант, В. Опдайк, Д. Робертс. – «Діалектика», 2018. – 448 с. 10. 16. Статичний та динамічний аналіз програмного забезпечення / Наукові записки НаУКМА. Комп'ютерні науки. Наукові записки НаУКМА. Комп'ютерні науки. / Режим доступу: <http://nrcscomp.ukma.edu.ua/article/view/220829>

13. Стрелковська І. В. Реінжиніринг та оптимізація програмних систем : методичні рекомендації для самостійної роботи здобувачів [Електронне видання] / І. В. Стрелковська, С. Б. Приходько, Т. І. Григор'єва ; кафедра інформаційних технологій Міжнародного гуманітарного університету. – Одеса, 2023. – 25 с. Режим доступу: <https://hdl.handle.net/11300/26663>

Додаткова:

1. Priyadarshi Tripathy, Kshirasagar Naik. Software Maintenance and Evolution: A Pattern-Oriented Approach. Режим доступу: <https://download.e-bookshelf.de/download/0002/8404/23/L-G-0002840423-0004532854.pdf>

2. Robert S. Arnold. Software Reengineering. Режим доступу: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication500-193.pdf>

3. Stanislav S. S. Velykodniy. The idealized models of software systems reengineering. Radio Electronics Computer Science Control. 2019. DOI:10.15588/1607-3274-2019-1-14

4. Velykodniy, S., Burlachenko, Z. i Zaitseva-Velykodna, S. (2019) Розробка архітектури програмного засобу для управління мережевим плануванням реінжинірингу програмного проекту. Сучасний стан наукових досліджень та технологій в промисловості, (2 (8), с. 25–35. doi: 10.30837/2522-9818.2019.8.025.

5. Великодний С. Реінжиніринг програмних систем та продуктів: Управління ІТ-проектами. 2019. 164 с.

6. Fung L., Tan S. Overview of Software Re-Engineering Concepts, Models and Approaches. JOIV: International Journal on Informatics Visualization. 2025. № 9. Pp. 46. DOI:10.62527/joiv.9.1.3034

7. Kernyskyy O., Kernyskyy A., Teslyuk V. The Synthesis Method for Specifications and Requirements in the Process of IT Project Reengineering. International Scientific and Technical Conference on Computer Sciences and Information Technologies, Institute of Electrical and Electronics Engineers Inc. 2023. № 5. Pp. 1–8. DOI:10.1109/CSIT61576.2023.10324175

8. Огнева О. С., Зярко Д. В. Дослідження ефективності моделей реінженерії програмного забезпечення для цифрової трансформації туристичного сервісу. Вісник Херсонського національного технічного університету. Том 2 № 3(94) (2025). <https://doi.org/10.35546/kntu2078-4481.2025.3.2.46>

9. Ian Sommerville Engineering Software Products: An Introduction to Modern Software Engineering, Global Edition. – Pearson. – 2020. – 368p.

10. Ситник Н. І., Каризька А. Р. Особливості реінжинірингу в організаціях, що займаються розробленням програмного продукту. / Режим доступу: <http://ape.fmm.kpi.ua/article/view/102773>

11. Глинчук Л.Я. Особливості процесу реінжинірингу програмного забезпечення. Стан, досягнення та перспективи інформаційних систем і технологій: матеріали XXIII Всеукраїнської наук.-техн. конф. молодих вчених, аспірантів та студентів. (Одеса, 20-21 квітня 2023 р.). Одеса: Видавництво ОНТУ, 2023. С. 218-220.

12. Михальчук Я.О., Гришанович Т.О., Глинчук Л.Я. Дослідження інтеграції месенджерів до вебсервісів. Прикладні проблеми комп'ютерних наук, безпеки та математики. 2023. № 1. С. 28-41. / Режим доступу: <https://apcssm.vnu.edu.ua/index.php/Journalone/article/view/8>
13. Глинчук Л.Я. Основні методи реінжинірингу програмних компонентів, їх складові та приклади застосування. Математика. Інформаційні технології. Освіта: збірник статей XII міжн. наук.-практ. конф. (Луцьк-Світязь, 3-5 червня 2023 р.). Луцьк: Волинський національний університет імені Лесі Українки, 2023. С.27-34.
14. Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall. 1st edition. 2019, 464p.
15. Robert C. Martin. Clean Agile: Back to Basics (Robert C. Martin Series) 1st Edition. Prentice Hall. 1st edition. 2019, 240p.
16. Robert C. Martin. Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series) 1st Edition. 2018, 318p.
17. Рефакторинг. Покращення існуючого коду. Мартін Фаулер Кент Бек. <https://leader-books.com.ua/kompyuterna-literatura/refactoring-pokrashennya-isnuyuchogo-kodu-martin-fauler-kent-bek>
18. Інтернет-портал для рефакторингу. URL: <https://refactoring.guru/uk>
19. Золотухіна О.А., Худік Б.О., Гавор А.С. Оптимізація продуктивності кешування за допомогою гібридного метода кешування даних. Зв'язок, 2025, №2. С. 78-86. <https://doi.org/10.31673/2412-9070.2025.027500>
20. 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>

ПОЛІТИКА КУРСУ («ПРАВИЛА ГРИ»)

Курс передбачає роботу в колективі.

- Середовище в аудиторії є дружнім, творчим, відкритим до конструктивної критики.
- Освоєння дисципліни передбачає обов'язкове відвідування лекцій і практичних занять, а також самостійну роботу.
- Самостійна робота включає в себе теоретичне вивчення питань, що стосуються тем лекційних занять, які не ввійшли в теоретичний курс, або ж були розглянуті коротко, їх поглиблена проробка за рекомендованою літературою.
- Усі завдання, передбачені програмою, мають бути виконані у встановлений термін.
- Якщо аспірант відсутній з поважної причини, він презентує виконання завдання під час самостійної підготовки та консультації викладача.
- Під час роботи над завданнями недопустимо порушення академічної доброчесності: при використанні Інтернет ресурсів та інших джерел інформації аспірант повинен вказати джерело, використане в ході виконання завдання. У разі виявлення факту плагіату аспірант отримує за завдання 0 балів.
- Аспірант, який спізнився, вважається таким, що пропустив заняття з неповажної причини з виставленням 0 балів за заняття, і при цьому має право бути присутнім на занятті.
- За використання телефонів і комп'ютерних засобів без дозволу викладача, порушення дисципліни аспірант видаляється з заняття, за заняття отримує 0 балів.

КРИТЕРІЇ ТА МЕТОДИ ОЦІНЮВАННЯ				
Умовою допуску до підсумкового контролю є набрання студентом 35 балів у сукупності за всіма темами дисципліни				
Форми контролю		Види навчальної роботи	Оцінювання	
ПОТОЧНИЙ КОНТРОЛЬ		Виконання практичних робіт	42 балів	
		Самостійна робота	18 балів	
ПІДСУМКОВЕ ОЦІНЮВАННЯ <i>Ispit</i>		Екзамен проходить у письмовій формі	40 балів	
ДОДАТКОВА ОЦІНКА				
Види навчальної роботи			Оцінювання, бали	
Участь у наукових конференціях, підготовка наукових публікацій за тематикою освітньої компоненти:				
• Тези доповіді на фаховій конференції			3	
• Стаття у фаховому виданні			5	
• Стаття в іноземному рецензованому виданні			10	
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	Аспірант 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 об'єму матеріалу, передбаченого робочою програмою, або студент проявляє невпевненість в тлумаченні теоретичних положень чи складних практичних завдань.		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)

82-89	Аспірант 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 / <i>Зараховано (B)</i>
75-81	Аспірант 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 / <i>Зараховано (C)</i>
67-74	Аспірант 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 / <i>Зараховано (D)</i>
60-66	Аспірант 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 / <i>Зараховано (E)</i>

35-59	Аспірант 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 відсутні.	Низький Не забезпечує практичної реалізації задач, що формуються при вивченні дисципліни	Незадовільно з можливістю повторного складання) / <i>Не зараховано (FX) В залікову книжку не проставляється</i>
0-34	Аспірант повністю не виконав вимог робочої програми навчальної дисципліни. Його знання на підсумкових етапах навчання є фрагментарними. Студент не допущений до перездачі іспиту.	Незадовільний Студент не підготовлений до самостійного вирішення задач, які окреслює мета та завдання дисципліни	Незадовільно з обов'язковим повторним вивченням / <i>Не допущений (F) В залікову книжку не проставляється</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.