

СИЛАБУС ОСВІТНЬОЇ КОМПОНЕНТИ «ПРИКЛАДНЕ ПРОГРАМУВАННЯ RUST, ч.2»

Лектор курсу			Миронюк Павло Ярославович, старший викладач.		Контактна інформація лектора (e-mail), сторінка курсу в Google Classroom	e-mail: p.myronuk@duikt.edu.ua; сторінка курсу в Google Classroom – https://classroom.google.com/c/ODM2Mjk2MzcwNzMy?cjc=iz3nzlvc	
Галузь знань					Рівень вищої освіти	бакалавр	
Спеціальність					Семестр		
Освітня програма					Тип дисципліни	Вибіркова	
Обсяг:	Кредитів ECTS	Годин	За видами занять:				
			Лекцій	Семінарських занять	Практичних занять	Лабораторних занять	Самостійна підготовка
	5	150	18	-	18	18	96

АНОТАЦІЯ КУРСУ

Взаємозв'язок у структурно-логічній схемі

Освітні компоненти, які передують вивченню	
Освітні компоненти для яких є базовою	

Мета курсу:	Формування знань та практичних навичок програмування мовою Rust, яка є сучасним інструментом для створення високопродуктивних та безпечних систем. Дисципліна спрямована на забезпечення глибокого розуміння концепцій управління пам'яттю, асинхронного та багатопотокового програмування, системи типів, механізмів, та гарантій безпеки, які закладені у Rust, а також їх застосування для створення надійного ПЗ.
--------------------	---

Компетентності відповідно до освітньої програми

Soft- skills / Загальні компетентності (ЗК)	Hard-skills / Спеціальні компетентності (СК)
Здатність застосовувати знання в практичних ситуаціях. Здатність вчитися та оволодівати сучасними знаннями. Здатність працювати в команді.	Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. Здатність розробляти архітектури, модулі та компоненти програмних систем. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення відповідно до вимог замовника, технічного завдання та стандартів. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення. Здатність до алгоритмічного та логічного мислення.

Програмні результати навчання (ПР)	
1. Знання	1.1. Назвати основні етапи становлення української державності.
2. Розуміння	2.1. Розуміти значення української державності для Європи.
3. Навички	3.1. Вміти аналізувати історичні документи.
4. Цінності	4.1. Цінувати українську державність.

Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.

Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.

ОРГАНІЗАЦІЯ НАВЧАННЯ			
Тема, опис теми	Вид заняття	Оцінювання за тему	Форми і методи навчання/питання до самостійної роботи
Тема 1. Система володінь та запозичень у Rust. Особливості системи типів Rust. <u>Знати:</u> Правила та принципи роботи системи типів, володінь та запозичень в Rust. Способи абстрагування поведінки. Відмінності, переваги, та недоліки різних типів поліморфізму в Rust. <u>Вміти:</u> Використовувати систему типів та гарантії компілятора Rust для проектування та створення ефективної та гнучкої архітектури ПЗ. Застосовувати найбільш оптимальні абстракції та методології створення ПЗ. <u>Рекомендовані джерела:</u> 1, 2, 3, 4, 5, 7, 9, 10.			
Заняття 1.1 Система володінь та запозичень. Семантика переміщення та копіювання.	Лекція 1 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 1.2. Гарантії Rust компілятора на практиці. Практичне застосування статичної типізації.	Практичне заняття 1 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 1.3. Оформлення Rust проекту та робота із менеджером залежностей на основі власного проекту.	Лабораторн е заняття 1 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Заняття 1.4 Динамічний та статичний поліморфізм. Відмінності та обмеження. Lifetimes.	Лекція 2 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 1.5 Використання поліморфізму для абстрагування поведінки.	Практичне заняття 2 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 1.6 Розширення поведінки альтернативними варіантами. Застосування динамічного поліморфізму на практиці.	Лабораторн е заняття 2 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Самостійна робота			
Способи обійти систему володінь у Rust.	8 год.	1 бал	Дослідити способи обійти системи володінь та запозичень у Rust у однопоточковому та багатопотокових контекстах.

Ергономічне конвертування даних між різними типами.	8 год.	1 бал	Дослідити методи зручного перетворення даних між різними типами. Дослідити трейти From/Into, TryFrom/TryInto, AsRef/AsMut, Borrow, Deref/DerefMut. Відмінності між наведеними трейтами.
Макроси та їх застосування.	8 год.	1 бал	Дослідити випадки використання та обмеження макросів у Rust. Навести приклади із проектів з відкритим кодом та Rust std. Відмінності процедурних та декларативних макросів.
Тема 2. Забезпечення коректності та стабільності ПЗ: системи багтрекінгу та робота із помилками. Знати: Способи, їх недоліки та переваги обробки помилок у Rust. Призначення та переваги використання багтрекінгових систем. Організація пріоритетів в середині проекту. Документування помилок та проекту. Вміти: Документувати проекти та помилки в Rust. Створювати відмовостійке на надійне ПЗ використовуючи переваги системи типів Rust. Інтегрувати багтрекінгові системи безпосередньо у процес розробки ПЗ. Рекомендовані джерела: 1, 2, 8.			
Заняття 2.1 Обробка та документування помилок в Rust.	Лекція 3 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 2.2. Відмінності реалізації помилок у Rust бібліотеках та застосунках.	Практичне заняття 3 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 2.3. Документування помилок та поведінки у Rust на проекті із попередньої лабораторної роботи.	Лабораторн е заняття 3 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Завдання 2.4 Багтрекінгові системи та їх інтеграція у проект. Організація пріоритетів.	Лекція 4 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Завдання 2.5 Порівняння різних багтрекінгових систем.	Практичне заняття 4 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Завдання 2.6 Інтеграція багтрекінгової системи у проект із попередньої лабораторної роботи.	Лабораторн е заняття 4 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Самостійна робота			
Локальне читання документації.	8 год.	1 бал	Навчитися використовувати Rust документацію та документацію залежностей локально. Дослідити методи

			локального кешування залежностей для використання у проєктах коли не має доступу до інтернету.
Можливості документування Rust проєктів.	8 год.	1 бал	Дослідити шляхи документування Rust проєктів: як створити власну документацію, який синтаксис написання Rust документації.
Публікація Rust бібліотеки на crates.io .	8 год.	1 бал	Навчитися публікувати Rust бібліотеки на crates.io .
Тема 3. Багатопотокові застосунки та гарантії що надає Rust. Знати: Небезпеки та складності розробки ПЗ у багатопотоковому контексті. Гарантії що Rust надає та не надає при роботі із потоками. Застосування Send та Sync трейтів для спільного доступу до даних. Вміти: Застосовувати поширені практики та систему типів Rust для уникнення гонок даних та взаємних блокувань. Використовувати бібліотеки екосистеми Rust для реалізації синхронізації та спільного доступу даних у багатопотоковому контексті. Рекомендовані джерела: 1, 2, 4, 7, 10.			
Заняття 3.1 Гарантії які Rust надає та не надає у багатопотоковому контексті.	Лекція 5 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 3.2. Передача даних між різними потоками та спільний доступ до даних.	Практичне заняття 5 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 3.3. Розпаралелення “важких” операцій на різних потоках у проєкті із попередньої лабораторної роботи.	Лабораторн е заняття 5 2 год	3 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Заняття 3.4 Синхронізація доступу до даних. Send та Sync трейти.	Лекція 6 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 3.5 Пошук частково синхронізованих типів у Rust std та їх аналіз.	Практичне заняття 6 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 3.6 Імплементация власних примітивів синхронізації.	Лабораторн е заняття 6 2 год	3 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Самостійна робота			
Проблеми багатопотокових застосунків, яким Rust не може запобігти.	8 год.	1 бал	Дослідити можливі проблеми багатопотокового виконання, такі як гонки ресурсів (не тільки даних) та взаємного блокування, методи діагностики та усунення цих проблем.
Неблокуючих структури даних в Rust.	8 год.	1 бал	Дослідити можливості неблокуючих структур даних у Rust, випадки їх використання та відмінності між їх імплементациями.

Використання каналів для полегшення роботи із потоками.	8 год.	2 бал	Дослідити та порівняти популярні бібліотеки в екосистемі Rust для роботи із каналами. Розглянути варіанти їх використання.
Тема 4. Асинхронне програмування в Rust. Знати: Призначення та відмінності асинхронного виконання від звичайного та багатопотокового. Можливості та обмеження асинхронного коду в Rust. Вміти: Будувати високопродуктивне ПЗ з використанням асинхронного програмування. Вміти використовувати поширені техніки та патерни для підвищення продуктивності, надійності, та організації асинхронного коду. Рекомендовані джерела: 1, 2, 6, 9.			
Заняття 4.1 Що таке асинхронне програмування. Відмінності асинхронного програмування у Rust від інших мов. Трейт Future.	Лекція 7 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 4.2. Імплементація примітивних асинхронних задач у Rust.	Практичне заняття 7 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 4.3. Підтримка асинхронного виконання у проєкті із попередньої лабораторної роботи.	Лабораторн е заняття 7 2 год	3 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Завдання 4.4 Особливості трейту Future: Pin та Unpin типи. Чому більшість асинхронних задач мають бути Send.	Лекція 8 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Завдання 4.5 Ручна імплементація типу Future.	Практичне заняття 8 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Завдання 4.6 Налаштування асинхронного рантайму для більшої продуктивності у проєкті із попередньої лабораторної роботи.	Лабораторн е заняття 8 2 год	3 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Завдання 4.7 Обмеження та патерни асинхронного програмування.	Лекція 9 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Завдання 4.8 Використання популярних бібліотек Rust для покращення асинхронного коду.	Практичне заняття 9 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Завдання 4.9 Рефакторинг та покращення асинхронного коду у проєкті із попередньої лабораторної роботи.	Лабораторн е заняття 9 2 год	3 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Самостійна робота			
Альтернативні асинхронні рантайми в Rust.	8 год.	1 бал	Дослідити можливі різних імплементацій асинхронних Rust рантаймів, окрім tokio.

Методи відміни (cancelling) асинхронних задач в Rust.	8 год.	2 бал	Дослідити способи відміни асинхронної задачі в Rust і чому це є складним завданням у Rust екосистемі.
Модель акторів для конкурентного виконання.	8 год.	2 бал	Дослідити модель акторів для конкурентного виконання задач. Порівняти із звичайним асинхронним виконанням. Знайти спільні та відмінні риси.
МАТЕРІАЛЬНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ			
• Мультимедійний проектор; • Комп'ютерний клас для проведення практичних занять. • Програмне забезпечення: Visual Studio Code із розширенням rust-analyzer або JetBrains RustRover (додаткові розширення не потрібні).			
ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ			
1. Martin Geisler. Comprehensive Rust. URL: https://google.github.io/comprehensive-rust/comprehensive-rust.pdf. 2. The Rust Programming Language: [Електронний ресурс]. – Режим доступу: https://doc.rust-lang.org/book/. 3. Rust By Example: [Електронний ресурс]. – Режим доступу: https://doc.rust-lang.org/stable/rust-by-example/. 4. Rustlings: [Електронний ресурс]. – Режим доступу: https://rustlings.rust-lang.org/. 5. The Little Book of Rust Macros: [Електронний ресурс]. – Режим доступу: https://lukaswirth.dev/tlborm/. 6. Asynchronous Programming in Rust: [Електронний ресурс]. – Режим доступу: https://rust-lang.github.io/async-book/. 7. Mara Bos. Rust Atomics and Locks: [Електронний ресурс]. – Режим доступу: https://marabos.nl/atomics/. 8. The rustdoc book: [Електронний ресурс]. – Режим доступу: https://doc.rust-lang.org/rustdoc/. 9. Rust Design Patterns: [Електронний ресурс]. – Режим доступу: https://rust-unofficial.github.io/patterns/. 10. Effective Rust: 35 Specific Ways to Improve Your Rust Code: [Електронний ресурс]. – Режим доступу: https://www.lurklurk.org/effective-rust/.			
ПОЛІТИКА КУРСУ («ПРАВИЛА ГРИ»)			
• 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 студент відсутній з поважної причини, він презентує виконані завдання під час самостійної підготовки та консультації викладача. • Під час роботи над завданнями не допустимо порушення академічної доброчесності: при використанні Інтернет ресурсів та інших джерел інформації студент повинен вказати джерело, використане в ході виконання завдання. У разі виявлення факту плагіату студент отримує за завдання 0 балів. • Студент, 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 студент видаляється з заняття, за заняття отримує 0 балів. • Код роботи студента має зберігатися у закритому GitHub репозиторії студента, доступ до якого має лише викладач та сам студент. • Роботи здаються у вигляді Pull Requests у приватний репозиторій. Критерієм прийняття роботи є approve на платформі GitHub. Якщо посилання завантажено у Classroom, а код відсутній – робота буде оцінена в 0 балів.			
КРИТЕРІЙ ТА МЕТОДИ ОЦІНЮВАННЯ			
Умовою допуску до підсумкового контролю є виконання всіх практичних робіт і виконання самостійних завдань, які передбачені структурою освітньої компоненти Інноваційне підприємництво та управління стартап проектами. Якщо студента не допущено до складання заліку, як такого, що не виконав індивідуальний план, йому надається час до перескладання для виконання всіх вимог допуску. Студент має право на два перескладання. При повторному перескладанні екзамену його у студента може приймати комісія, яка створюється директором ННІТ. Оцінка комісії є остаточною. У випадку отримання студентом 0 балів (неприйнятно), що тягне відрахування за невиконання навчального плану.			

Оцінювання студентів здійснюється за накопичувальною 100-бальною системою і складається із двох основних оцінкових блоків і розподіляється в певних пропорціях 60 (бали напрацьовані під час вивчення дисципліни – Поточний контроль), 40 (підсумкове оцінювання - Екзамен):			
Форми контролю	Види навчальної роботи		Оцінювання
ПОТОЧНИЙ КОНТРОЛЬ	● Виконання практичних робіт		45 балів
	● Самостійна робота		15 балів
ПІДСУМКОВЕ ОЦІНЮВАННЯ Екзамен	Залік проходить в письмовій формі.		40 балів
Додаткова оцінка			
Види навчальної роботи			Оцінювання
Проходження Rustcamp від Української Rust спільноти (https://github.com/rust-lang-ua/rustcamp): Якщо студент уже проходив цей курс і по ньому були нараховані додаткові бали, то другий раз бали не зараховуються.			
- Завершення базової частини Rustcamp. Студент повинен мати сертифікат про закінчення курсу та успішно складений екзаменаційний тест.			5 балів
- Завершення базової частини Rustcamp + web спеціалізація. Студент повинен мати сертифікат про закінчення курсу та успішно складений екзаменаційний тест.			10 балів
Проект написаний мовою програмування Rust:			
- Проект включає роботу із файлами/мережею/базою даних, асинхронне виконання, налаштоване логування, гнучка обробка помилок та містить проектну документацію.			5 балів
- Проект включає роботу із файлами/мережею/базою даних, асинхронне виконання, налаштоване логування, гнучка обробка помилок та містить проектну документацію. Налаштоване середовище CI/CD, більшість логіки покрита тестами, основна поведінка винесена в окремий крейт, використання патернів Rust.			10 балів
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		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)			

	виходить за рамки 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) В залікову книжку не представляється
1-34	Студент повністю не виконав вимог робочої програми навчальної дисципліни. Його знання на підсумкових етапах навчання є фрагментарними.	Незадовільний Студент не підготовлений до	Незадовільно з обов'язковим повторним вивченням /

	Студент не допущений до здачі екзамену/заліку.	самостійного вирішення задач, які окреслює мета та завдання дисципліни	Не допущений (F) <i>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.