ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ РІК ЗАСНУВАННЯ - 1966

Відділення інформаційних технологій

ЦК програмної інженерії та інтелектуальних систем

СИЛАБУС

Базова інформація про дисципліну	
Назва дисципліни	SE051 Розробка програмного забезпечення за допомогою штучного інтелекту / AI-Assisted Software Development
Рівень фахової передвищої освіти	Фаховий молодший бакалавр
Галузь знань	F «Інформаційні технології»
Спеціальність	F2 «Інженерія програмного забезпечення»
Освітня програма	Інженерія програмного забезпечення
Семестр	5 (на базі базової середньої освіти) 3 (на базі повної загальної середньої освіти)
Курс	2 (на базі повної загальної середньої освіти) 3 (на базі базової середньої освіти)
Анотація курсу	Дисципліна присвячена інженерії розробки програмного забезпечення в умовах повсякденного застосування генеративних інструментів штучного інтелекту: чат-асистентів, асистентів коду та агентних середовищ. Курс побудований навколо наскрізного індивідуального проєкту, що проходить усі стадії життєвого циклу: від специфікації з перевірними критеріями приймання, карти екранів і моделі даних до серверної частини з розмежуванням доступу, функції на базі мовної моделі, конвеєра безперервної інтеграції та публічного розгортання.
Сторінка курсу в MOODLE	http://78.137.2.119:2929/course/view.php?id=999
Мова викладання	українська
Лектор курсу	Лось Владислав Юрійович, викладач Канали комунікації: E-mail: vladlos.edu@gmail.com; Чат на сторінці дисципліни в СДН «Moodle»
Місце дисципліни в освітній програмі	
Освітньо-професійна програма	http://csbc.edu.ua/documents/otdel/moop_pr24.pdf
Перелік загальних компетентностей (ЗК)	**ЗК:** - Знання та розуміння предметної області та розуміння професійної діяльності. - Здатність до пошуку, оброблення та аналізу інформації з різних джерел. - Здатність застосовувати знання в практичних ситуаціях. - Здатність дотримуватися принципів академічної доброчесності та нести відповідальність за результати власної інтелектуальної праці.

Перелік спеціальних компетентностей (СК)	**СК:** - Здатність алгоритмічно та логічно мислити. - Здатність вдосконалювати знання і навички в галузі інформаційних технологій та усвідомлення важливості навчання протягом усього життя. - Здатність застосовувати теоретичні та емпіричні знання для розроблення, тестування, впровадження та супроводу програмного забезпечення. - Здатність дотримуватися стандартів при розробці програмного забезпечення. - Здатність брати участь у визначенні та формулюванні вимог до програмного забезпечення. - Здатність брати участь у проектуванні програмного забезпечення. - Здатність вибирати та використовувати ефективні інструментальні засоби розробки програмного продукту. - Здатність використовувати методи та інструменти штучного інтелекту для автоматизації процесів розробки, налаштування та діагностики програмного забезпечення.
Перелік програмних результатів навчання	**РН:** - Застосовувати основні принципи професійної етики у галузі програмної інженерії, усвідомлювати їх соціальну значимість та культурні аспекти в професійній діяльності. - Систематизувати та узагальнювати інформацію про підходи, методи та засоби розробки та супроводу програмного забезпечення. - Застосовувати спеціалізовані емпіричні та теоретичні знання у сфері інженерії програмного забезпечення. - Використовувати знання математичних методів на рівні, необхідному для розв’язання типових задач програмної інженерії. - Розробляти та супроводжувати програмне забезпечення. - Використовувати основні методології та підходи до організації життєвого циклу програмного забезпечення. - Застосовувати стандарти, специфікації в процесах життєвого циклу програмного забезпечення. - Аналізувати вимоги до програмного забезпечення. - Обирати інструментальні засоби, ефективні методи та здійснювати тестування програмних систем. - Розуміти предметну область, застосовувати знання у професійній діяльності. - Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для розв’язання професійних задач з урахуванням сучасних досягнень інформаційних технологій.

	- Виконувати розгортання та налаштування програмного забезпечення та засобів захисту відповідно до політики інформаційної безпеки.
Опис дисципліни	
Структура навантаження на студента	Загальна кількість годин – 90 Кількість кредитів – 3 Кількість лекційних годин – 30 Кількість практичних занять – 30 Кількість годин для самостійної роботи студентів – 30 Форма підсумкового контролю – диференційований залік
Методи навчання	Словесні (розповідь, пояснення, обговорення ТЗ, дискусія, інструктаж); наочні (презентаційні повідомлення); практичні роботи; інтерактивні методи (дистанційні консультації); індивідуальні завдання
Зміст дисципліни	
Змістовий модуль 1. Підготовка розробки: промпти, вимоги, архітектура	
Тема 1. Життєвий цикл розробки програмного забезпечення і місце штучного інтелекту в ньому	Життєвий цикл програмного забезпечення за ISO/IEC/IEEE 12207. Роль генеративних інструментів і типова пастка на кожному етапі. Принцип роботи великої мовної моделі: авторегресивна генерація, зріз знань, безстановість, недетермінованість, галюцинації. Режими роботи інструментів асистованої розробки. Керування контекстом генерації.
Тема 2. Інженерія промптів як інженерія специфікацій	Промпт як вимога до результату. Компоненти технічного промпту. Стратегії подання задачі. Сигнатура, контракт, схема даних і таблиця прикладів як засоби специфікації. Розмір кроку. Типи галюцинацій і прийоми зниження їх імовірності.
Тема 3. Vibe coding: швидкість прототипування та її ціна	Означення vibe coding. Межі доречного застосування стилю. Механізми накопичення технічного боргу: дублювання коду, неузгодженість абстракцій, розростання залежностей. Критерії переходу «прототип → продукт». Журнал взаємодії як вимірювання ціни розробки без підготовки.
Тема 4. Інженерія вимог із залученням генеративних моделей	Вимога за ISO/IEC/IEEE 29148. Функціональні, нефункціональні вимоги та обмеження. Користувацька історія, критерії приймання, сценарій Given/When/Then. Вигадана вимога і процедура маркування джерела. Пріоритезація MoSCoW, фіче-план і межі MVP. Модель як опонент власного плану.

Тема 5. Архітектурне проєктування та моделювання даних	Монолітна та мікросервісна архітектура. Шарова організація коду й напрямок залежностей, ізоляція шару доступу до даних. Схема даних від потреб екранів: сутності, зв'язки, третя нормальна форма, індекси. Міграції як єдине джерело схеми. Мінімальний контракт API. Архітектурне рішення у форматі ADR.
Тема 6. Робота з наявною кодовою базою	Процедура входження в незнайомий проєкт. Контекстне вікно й техніки роботи з великим репозиторієм. Реверс-документування та його перевірка. Характерні збої агента на наявному коді. Характеризаційний тест.
Практичні роботи	ПР 1. Vibe coding: прототип без підготовки. ПР 2. Ідея, фіче-план і специфікація проєкту. ПР 3. Структура інтерфейсу застосунку. Модульна контрольна робота № 1.
Змістовий модуль 2. Реалізація, якість, безпека та постачання програмного продукту	
Тема 7. Клієнтська частина застосунку	Компонент інтерфейсу. Презентаційні та контейнерні компоненти. Джерела істини для стану застосунку. Чотири стани екрана. Клієнтська і серверна валідація форм. Мінімальні перевірки доступності.
Тема 8. Серверна частина та персистентність	Маршрут, валідація, сервіс, репозиторій. Валідація на межі системи. Об'єктно-реляційне відображення та прямі запити, проблема N+1. Автентифікація й авторизація на рівні ресурсу, політики доступу на рівні рядка. Секрети у змінних середовища, публічний і службовий ключі.
Тема 9. Версійний контроль, командна робота та рецензування машинного коду	Модель «гілка на задачу», атомарний коміт, формат повідомлень комітів. Запит на злиття й обсяг зміни. Пріоритетний порядок рецензування машинного коду. Конфлікт злиття. Декларування використання асистента. Файл фіксації версій і перевірка нових залежностей.
Тема 10. Інтеграція інтелектуальної функції у застосунок	Виклик мовної моделі з серверного коду. Структурований вивід: схема, парсинг, валідація, повторна спроба. Пошук із розширеним контекстом. Агентні сценарії. Ін'єкція промпту та рубежі захисту: ліміт частоти запитів, тайм-аут, бюджет витрат, кешування.
Тема 11. Тестування, налагодження та розробка через тестування	Структура тесту й тестова піраміда. Чутливість тесту та мутаційне тестування. Тавтологічні тести. Послідовність налагодження від стектрейса до підтвердженого виправлення. Регресійний тест. Цикл розробки через тестування, правило «тест перший».
Тема 12. Рефакторинг і керування технічним боргом	Означення рефакторингу та передумова тестового покриття. Запахи коду: дублювання, довгі функції, зайві параметри, передчасна абстракція. Межі принципів DRY та єдиної відповідальності. Реєстр технічного боргу. Поділ великого рефакторингу на кроки.

Тема 13. Безпека машинно згенерованого коду	Типові вразливості машинно згенерованого коду: порушення контролю доступу, ін’єкції, криптографічні збої, некоректна конфігурація. Загрози застосунків на базі мовних моделей. Слопсквотинг. Секрети та ротація після витоку. Статичний аналіз і аудит залежностей.
Тема 14. Безперервна інтеграція, контейнеризація та розгортання	Відтворюваність збірки. Образ, шар образу, багатоетапна збірка. Конфігурація через середовище. Конвеєр безперервної інтеграції та блокування злиття. Міграції відносно розгортання, відкат розгортання. Журнал застосунку.
Тема 15. Право, етика та відповідальність	Ліцензії відкритого коду. Дані у запитах до сторонніх сервісів. Відповідальність за дефект незалежно від авторства коду. Обчислювальна вартість інференсу. Декларація використання інструментів штучного інтелекту.
Практичні роботи	ПР 4. Модель даних під інтерфейс. ПР 5. Серверна частина, доступ і зшивання з інтерфейсом. ПР 6. Стабільність і безпека функції на базі мовної моделі. ПР 7. Безпека, конвеєр і продакшн. Модульна контрольна робота № 2.
Політика дисципліни	
Політика відвідування	Регулярне відвідування всіх видів занять, своєчасність виконання самостійної роботи. За об’єктивних причин студент може отримати індивідуальний графік навчання за погодженням із керівником курсу, завідувачем ЦК та завідувачем відділення. Практичні роботи № 2–7 виконуються індивідуально на одному наскрізному проєкті й лише послідовно: результат кожної роботи є вхідними даними наступної.
Політика щодо дедлайнів та перескладання	Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Допустиме відставання — не більше однієї практичної роботи. Перескладання модульної контрольної роботи здійснюється у строк, погоджений із викладачем, у межах семестру.
Академічна доброчесність	У випадку недотримання політики академічної доброчесності (плагіат, самоплагіат, фабрикація, фальсифікація, списування, обман, хабарництво) передбачено повторне проходження оцінювання. Порушенням вважається присвоєння результату роботи моделі за відсутності здатності пояснити його внутрішню логіку, а не сам факт генерації. Самостійність підтверджується власним репозиторієм з історією комітів, повним експортом діалогу з моделлю з наскрізною нумерацією звернень та усним захистом практичних робіт.

Використання ШІ	Використання ШІ під час виконання завдань регламентується Політикою «Використання ШІ в освітньому процесі ЧДБК». Завдання мають маркування регламенту використання ШІ. У межах дисципліни використання генеративних інструментів (Google AI Studio, GitHub Copilot) дозволене і обов'язково декларується в журналі взаємодії кожної роботи. Використання асистента заборонене під час модульних контрольних робіт та усних відповідей на захисті практичних робіт. Ключі доступу, паролі та персональні дані асистентам не передаються.
Підсумковий контроль	Диференційований залік у кінці семестру за результатами поточної успішності (у формі підсумкового модульного контролю)

Система оцінювання

Система оцінювання підсумкової успішності студентів поділяється на **поточний контроль** та **семестровий контроль**.

Поточний контроль здійснюється протягом семестру і охоплює всі види аудиторної роботи (практичні заняття, модульні контрольні роботи) та виконання індивідуального завдання (ІНДЗ). Максимальна кількість балів, яку студент може набрати за цей вид контролю, становить 100.

Підсумковий контроль

Відбувається у формі диференційованого заліку (у формі підсумкового модульного контролю).

Розрахунок підсумкової оцінки

Для предметів із заліком: Підсумкова оцінка базується виключно на балах, накопичених протягом семестру (**S**). Ваговий коефіцієнт у цьому випадку становить **1**.

Формула: O=S×1

Види навчальної роботи	**Загальна кількість балів**
Практичні роботи № 1–7 (ПР № 1 – 5 б.; ПР № 2–4 – по 7 б.; ПР № 5–7 – по 8 б.)	50
Модульні контрольні (2 к.р. по 10 балів)	20
Індивідуальне завдання (ІНДЗ)	30
Разом	**100**

Критерії оцінювання для кожного виду навчальної роботи

Критерії оцінювання практичних робіт

Максимальний бал за практичну роботу залежить від її складності: ПР № 1 – 5 б.; ПР № 2–4 – по 7 б.; ПР № 5–7 – по 8 б. Оцінка виставляється як частка від максимального бала з округленням до цілого:

100% – робота здана у встановлений строк, усі кроки методичних вказівок виконано; результат відтворюється (репозиторій, тег здачі, для ПР 2–7 — публічне посилання); звіт відповідає єдиному формату, журнал взаємодії збігається з експортом діалогу; тести мають підтверджений червоний стан (з ПР 3); студент упевнено пояснює логіку будь-якого фрагмента коду на усному захисті роботи.

80–90% – робота виконана повністю, наявні незначні недоліки у звіті, журналі взаємодії або якості коду; відповіді на захисті містять несуттєві неточності.

60–70% – робота містить помилки, що не перешкоджають досягненню мети, або здана з відставанням на одну роботу без поважної причини.

40–50% – виконано основну частину кроків; окремі артефакти (тести, журнал взаємодії, скріншоти) неповні; незавершені кроки описані у звіті з причиною; студент пояснює результат частково.

20–30% – результат відтворюється частково, у звіті бракує ключових артефактів.

10% – роботу подано формально: результат не відтворюється або студент не пояснює логіку поданого коду й власних рішень.
0 б. – роботу не подано.
У ПР 1 якість коду не оцінюється: оцінюються повнота журналу взаємодії та історії комітів, аудит розуміння і відповіді на захисті. Незавершений крок із чесним описом причини оцінюється вище, ніж прихована прогалина.

Критерії оцінювання модульних контрольних робіт (10 б. за кожну роботу)
Модульна контрольна робота № 1 охоплює змістовий модуль 1 (теми 1–6), № 2 — змістовий модуль 2 (теми 7–14). Кожна робота містить теоретичну частину (тестові та відкриті питання) і практичну частину — аналіз наданого фрагмента машинно згенерованого коду або специфікації. Робота виконується без застосування генеративних інструментів.
10 б. – теоретичні питання розкрито повно й термінологічно точно; у практичній частині самостійно виявлено всі дефекти, встановлено їх причини та обґрунтовано виправлення.
8–9 б. – теоретичні питання розкрито повно; у практичній частині виявлено основні дефекти, допущено окремі неточності в обґрунтуванні.
6–7 б. – матеріал відтворено загалом правильно, але без деталізації; практичну частину виконано частково.
4–5 б. – відповіді фрагментарні; у практичній частині виявлено окремі дефекти без пояснення причин.
1–3 б. – студент відтворює лише окремі означення; практичну частину не виконано.
0 б. – роботу не виконано або не здано у встановлений строк без поважної причини.

Критерії оцінювання індивідуального завдання (30 б.)
Індивідуальне завдання (ІНДЗ) охоплює частину завдань практичних робіт, виконується в межах наскрізного проєкту та оцінюється сумарно:
27–30 б. – завдання виконано повністю, аналіз глибокий, використано додаткові джерела; робота структурована, логічна, без помилок; оформлення відповідає вимогам.
24–26 б. – виконано 85–95% завдань; є незначні помилки або неточності, але матеріал розкрито достатньо глибоко; структура відповідає вимогам.
21–23 б. – виконано 70–85% завдань; є помилки в аналізі або прикладах, але основні положення засвоєно; оформлення з дрібними недоліками.
18–20 б. – виконано 60–70% завдань; тему розкрито частково, бракує аналізу або власних висновків; є суттєві помилки.
15–17 б. – виконано 50–60% завдань; відповіді поверхневі, немає посилань на джерела; слабка структура.
10–14 б. – виконано 30–50% завдань; численні помилки, тему майже не розкрито; оформлення не відповідає вимогам.
1–9 б. – виконано лише окремі завдання (до 30%); немає висновків, робота фрагментарна.
0 б. – роботу не здано або вона не містить змістовних відповідей.

ШКАЛА ОЦІНЮВАННЯ ЗНАНЬ ТА ВМІНЬ ЗДОБУВАЧІВ ОСВІТИ

Бали	Зміст
90-100 відмінно	**Бездоганна підготовка в широкому контексті.** Здобувач виявляє глибокі, системні знання навчального матеріалу, вільно орієнтується в першоджерелах та додатковій літературі, демонструє творчий підхід, аналітичне мислення та здатність застосовувати набуті компетентності в нестандартних ситуаціях.
80-89 дуже добре	**Повні знання, міцні вміння.** Здобувач володіє навчальним матеріалом у повному обсязі, впевнено застосовує практичні навички, вільно виконує передбачені програмою завдання, проте припускається несуттєвих неточностей або незначних помилок при викладенні окремих питань
70-79 добре	**Хороші знання та вміння.** Здобувач грамотно та по суті викладає матеріал, демонструє достатній рівень засвоєння практичних навичок, але

	припускається деяких неточностей у формулюваннях або має невеликі труднощі під час самостійного виконання складних завдань.
65-69 задовільно	**Задовільні знання, стереотипні вміння.** Здобувач знає основний навчальний матеріал в обсязі, необхідному для подальшого навчання, вирішує завдання за готовим алгоритмом (шаблоном), проте має прогалини в деталях та припускається помилок при самостійних висновках.
60-64 достатньо	**Виконання мінімальних вимог діяльності в стандартних умовах.** Здобувач демонструє засвоєння лише базового рівня матеріалу, спроможний виконувати лише найпростіші завдання в стандартних (знайомих) умовах під керівництвом викладача.
35-59 незадовільно	**Слабкі знання, відсутність вмінь.** Здобувач володіє лише фрагментарними знаннями, не здатний системно викласти матеріал чи застосувати вміння на практиці. Потрібне повторне складання (перескладання) для досягнення мінімального рівня.
1-34 незадовільно	**Необхідний повторний курс.** Здобувач демонструє повну відсутність необхідних знань та вмінь, що унеможливлює складання заборгованості без повторного вивчення навчальної дисципліни

Список рекомендованих джерел

1. *Beck K.* Tidy First? : A Personal Exercise in Empirical Software Design. Sebastopol : O'Reilly Media, 2023. 122 p.
2. *Osmani A.* Beyond Vibe Coding : From Coder to AI-Era Developer. Sebastopol : O'Reilly Media, 2025. 252 p.
3. *Martin R. C.* Clean Code : A Handbook of Agile Software Craftsmanship. 2nd ed. Boston : Addison-Wesley, 2025. 672 p.
4. *Aniche M.* Effective Software Testing : A Developer's Guide. Shelter Island : Manning, 2022. 328 p.
5. Software Architecture : The Hard Parts / N. Ford, M. Richards, P. Sadalage, Z. Dehghani. Sebastopol : O'Reilly Media, 2021. 459 p.
6. DORA 2025 State of AI-assisted Software Development Report / D. DeBellis et al. Google Cloud, 2025. URL: https://research.google/pubs/dora-2025-state-of-ai-assisted-software-development-report/ (дата звернення: 05.10.2026).
7. *Wiegers K., Hokanson C.* Software Requirements Essentials : Core Practices for Successful Business Analysis. Boston : Addison-Wesley, 2023. 208 p.
8. *Skoulikari A.* Learning Git : A Hands-On and Visual Guide to the Basics of Git. Sebastopol : O'Reilly Media, 2023. 320 p.
9. ISO/IEC/IEEE 12207:2026. Systems and software engineering. Software life cycle processes. [Чинний від 2026-04]. 2nd ed. Geneva : ISO, 2026. 140 p.
10. ISO/IEC/IEEE 29148:2018. Systems and software engineering. Life cycle processes. Requirements engineering. [Чинний від 2018-11, підтверджений у 2024]. 2nd ed. Geneva : ISO, 2018. 92 p.
11. Про авторське право і суміжні права : Закон України від 01.12.2022 № 2811-IX. URL: https://zakon.rada.gov.ua/laws/show/2811-20 (дата звернення: 05.10.2026).
12. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI. URL: https://zakon.rada.gov.ua/laws/show/2297-17 (дата звернення: 05.10.2026).

Інтернет-ресурси

1. OWASP Top 10. URL: https://owasp.org/Top10/ (дата звернення: 05.10.2026).
2. OWASP Top 10 for LLM Applications 2025. URL: https://genai.owasp.org/llm-top-10/ (дата звернення: 05.10.2026).

3. OWASP Cheat Sheet Series. URL: https://cheatsheetseries.owasp.org/ (дата звернення: 05.10.2026).
4. Next.js Documentation. URL: https://nextjs.org/docs (дата звернення: 05.10.2026).
5. Supabase Documentation. URL: https://supabase.com/docs (дата звернення: 05.10.2026).
6. Vitest Documentation. URL: https://vitest.dev/ (дата звернення: 05.10.2026).
7. GitHub Docs: Codespaces. URL: https://docs.github.com/en/codespaces (дата звернення: 05.10.2026).
8. GitHub Docs: Actions. URL: https://docs.github.com/en/actions (дата звернення: 05.10.2026).
9. Vercel Documentation. URL: https://vercel.com/docs (дата звернення: 05.10.2026).
10. Web Content Accessibility Guidelines (WCAG) 2.2. URL: https://www.w3.org/TR/WCAG22/ (дата звернення: 05.10.2026).
11. Шаблонний репозиторій курсу. URL: https://github.com/vladlos/AIDEVCOURSE (дата звернення: 05.10.2026).