

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Кафедра вищої математики та інформатики

“ЗАТВЕРДЖУЮ”

В. о. декана факультету математики і інформатики

Свген МЕНЯЙЛОВ



31 ” серпня 2026 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ ІНФОРМАТИКА ТА ПРОГРАМУВАННЯ

рівень вищої освіти перший (бакалаврський) рівень

галузь знань A Освіта

спеціальність A4.04 Середня освіта (Математика)

освітня програма Математика та інформатика

спеціалізація -

вид дисципліни обов'язкова

факультет математики і інформатики

2026 / 2027 навчальний рік

Програму рекомендовано до затвердження вченою радою факультету математики і інформатики

“ 28 ” серпня 2026 року, протокол № 8

РОЗРОБНИК ПРОГРАМИ: Олександр ДЕЙНЕГА, доктор філософії з комп'ютерних наук, викладач закладу вищої освіти кафедри вищої математики та інформатики.

Програму схвалено на засіданні кафедри вищої математики та інформатики

Протокол від “26” серпня 2026 року № 1

В. о. завідувача кафедри [підпис] Ірина ЖОВТОНІЖКО

Програму погоджено з гарантом освітньо-професійної програми

«Математика та інформатика»

Гарант освітньо-професійної програми А. Черн Ганна ЧЕРНОВА

Програму погоджено науково-методичною комісією факультету математики і інформатики

Протокол від “27” серпня 2026 року № 1

Голова науково-методичної комісії [підпис] Євген МЕНЯЙЛОВ

ВСТУП

Програма навчальної дисципліни «Інформатика та програмування» складена відповідно до освітньо-професійної програми «Математика та інформатика» першого (бакалаврського) рівня вищої освіти спеціальності (предметної спеціальності) А4.04 Середня освіта (Математика).

1. Опис навчальної дисципліни

1.1. Мета викладання навчальної дисципліни

Мета викладання навчальної дисципліни «Інформатика та програмування» полягає у формуванні системних знань з алгоритмізації, програмування та структур даних, розвитку здатності розробляти, аналізувати, тестувати й налагоджувати програмні рішення мовами Python і C, а також застосовувати алгоритмічні підходи та програмні засоби для розв'язання інформатичних, математичних і прикладних задач.

1.2. Основні завдання вивчення дисципліни

- сформувати розуміння базових понять алгоритмізації та принципів побудови програм;
- оволодіти основними конструкціями, типами даних і засобами структурованого програмування мовами Python та C;
- набути навичок процедурного та об'єктно-орієнтованого програмування мовою Python;
- сформувати вміння тестувати, налагоджувати програми, обробляти винятки та працювати з файлами;
- оволодіти засобами мови C для роботи з масивами, функціями, вказівниками, динамічною пам'яттю, рядками, структурами та файлами;
- сформувати навички побудови та використання зв'язаних структур даних, дерев, стеків, черг, множин, словників і графів;
- набути здатності аналізувати коректність і складність алгоритмів, порівнювати способи їх реалізації;
- опанувати базові та удосконалені алгоритми внутрішнього і зовнішнього сортування;
- розвинути навички створення цифрових продуктів і програмних моделей для розв'язання задач математичної та інформатичної освітніх галузей.

1.3. Кількість кредитів

15 кредитів ЄКТС.

1.4. Загальна кількість годин

450 годин.

1.5. Характеристика навчальної дисципліни

Семе стр	Рік підгото вки	Кред ити	Усього годин	Лекці ї	Практ ичні	Лабора торні	Самостій на робота	Індивідуальні завдання	Підсумков ий контроль
1	1-й	4	120	32	16	0	72	1 розрахунково- графічна робота*	залік

Семе- стр	Рік підгото- вки	Кред- ити	Усього годин	Лекці- ї	Практ- ичні	Лабора- торні	Самостій- на робота	Індивідуальні завдання	Підсумков- ий контроль
2	1-й	6	180	32	32	0	116	3 розрахунково- графічні роботи*	екзамен
3	2-й	5	150	0	16	0	134	3 розрахунково- графічні роботи*	екзамен
Разо- м	-	15	450	64	64	0	322	7 розрахунково- графічних робіт*	-

* Розрахунково-графічні роботи виконуються в межах годин самостійної роботи, передбачених навчальним планом.

Заочна (дистанційна) форма здобуття освіти освітньо-професійною програмою не передбачена.

1.6. Перелік компетентностей, що формує дана дисципліна

Інтегральна компетентність

ІК01. Здатність успішно розв'язувати складні спеціалізовані професійно-педагогічні задачі та проблеми математичної та інформатичної освітніх галузей при роботі зі здобувачами загальної середньої освіти, що передбачає застосування теоретичних та практичних знань, психолого-педагогічних теорій та фахових методик, сучасних педагогічних технологій та характеризується комплексністю та невизначеністю умов в освітньому середовищі.

Загальні компетентності

ЗК03. Здатність вчитися та оволодівати сучасними знаннями, використовувати їх у практичних ситуаціях в освітній діяльності, мислити абстрактно, здійснювати аналіз і синтез, володіти культурою мислення.

ЗК05. Здатність знаходити, обробляти та аналізувати інформацію для розвитку сучасного суспільства дотримуючись вимог інформаційної безпеки.

ЗК06. Здатність до креативності та генерування нових ідей.

Фахові компетентності

ФК03. Інформаційно-цифрова компетентність, що описує здатність орієнтуватися в інформаційному просторі, критично оцінювати та оперувати інформацією в професійній діяльності, ефективно використовувати наявні та створювати нові електронні ресурси, а також застосовувати цифрові технології при навчанні математичної та інформатичної освітнім галузям в умовах невизначеності.

ФК14. Інформатична компетентність, що передбачає здатність раціонально розв'язувати інформатичні задачі різного рівня складності, використовуючи програмні засоби загального та спеціального призначення, створювати цифрові продукти для розв'язання задач математичної та інформатичної освітніх галузей.

1.7. Перелік результатів навчання, що формує дана дисципліна

ПРН03. Орієнтуватися в інформаційному просторі, критично оцінювати та використовувати інформацію, демонструючи медійно-інформаційну грамотність та цифрову компетентність для ефективного

вирішення професійних завдань при навчанні математичної та інформатичної галузям, зокрема в умовах невизначеності.

ПРН18. Розв'язувати інформатичні задачі різного рівня складності раціональними методами з використанням програмних засобів загального та спеціального призначення.

ПРН19. Створювати цифрові продукти для розв'язання задач математичної та інформатичної освітніх галузей за допомогою цифрових пристроїв, будувати інформаційні моделі засобами цифрових технологій, презентувати отримані результати.

1.8. Пререквізити

Базові знання зі шкільного курсу інформатики та математики; навички роботи з персональним комп'ютером і базовими програмними засобами.

2. Тематичний план навчальної дисципліни

Розділ 1. Початкові відомості про алгоритми і програми. Мова програмування Python

Тема 1. Алгоритми та їх основні структури

Поняття алгоритму. Послідовність виконання дій, розгалуження та цикли як базові структури алгоритмів. Використання блок-схем для подання алгоритмів. Псевдокод як спосіб опису. Види циклів (з передумовою, з післяумовою, з лічильником). Реалізація найпростіших алгоритмів мовою Python.

Тема 2. Типи даних

Архітектура комп'ютера та способи зберігання інформації в пам'яті. Основні типи даних Python: цілі та дійсні числа, логічні значення, рядки, списки, словники, множини. Операції та перетворення типів.

Тема 3. Процедурне програмування

Метод покрокової деталізації під час розробки програм. Функції, параметри та аргументи. Модулі. Вкладені виклики. Основи рекурсії. Структурування програмного коду.

Тема 4. Тестування, налагодження, винятки. Робота з файлами

Основи налагодження. Використання конструкцій try/except. Перевірка правильності за допомогою assert. Файлове введення/виведення. Обробка текстових файлів. CSV та JSON.

Тема 5. Аналіз алгоритмів

Поняття коректності алгоритмів та методи її доведення. Складність алгоритмів: основні поняття й підходи до оцінки. Класичні алгоритми пошуку та сортування.

Тема 6. Основи об'єктно-орієнтованого програмування в Python

Класи, об'єкти, атрибути та методи. Наслідування, перевизначення методів, поліморфізм.

Розділ 2. Загальні поняття мови програмування C

Тема 7. Базові елементи мови C

Компілятори та інтегровані середовища розробки. Структура програми мовою C. Директиви препроцесора. Стандартні бібліотеки. Базовий синтаксис.

Тема 8. Типи даних мови C

Концепція типу даних. Класифікація типів даних. Базові типи та їх машинне представлення. Змінні та константи. Введення та виведення даних засобами бібліотеки stdio.

Розділ 3. Операції, оператори та масиви

Тема 9. Операції та їх пріоритети

Арифметичні, логічні, порівняльні та інші операції мови C. Пріоритет і асоціативність операцій. Побітові операції та їх застосування.

Тема 10. Оператори мови C

Умовні оператори if, else if, else. Багатоваріантний вибір switch. Цикли for, while, do...while. Використання операторів break і continue.

Тема 11. Масиви

Одновимірні та багатовимірні масиви. Оголошення та ініціалізація. Індекссування та обхід масивів. Використання масивів для реалізації операцій над матрицями.

Розділ 4. Функції у мові C

Тема 12. Призначення, означення і використання функцій

Прототип, визначення та виклик функції. Параметри й аргументи. Покрокова деталізація програм. Передавання даних у функції.

Тема 13. Область видимості та класи зберігання

Глобальні та локальні змінні. Область видимості та тривалість зберігання. Класи зберігання. Принцип локалізації. Основи багатофайлових проєктів.

Розділ 5. Вказівники та динамічні змінні

Тема 14. Поняття вказівника

Адреси пам'яті. Операції отримання адреси та розіменування. Оголошення та використання вказівників. Вказівники як параметри функцій.

Тема 15. Вказівники і масиви

Взаємозв'язок масивів і вказівників. Арифметика вказівників. Динамічні масиви. Динамічне виділення пам'яті. Способи зберігання двовимірних динамічних масивів.

Розділ 6. Складені типи даних та робота з файлами

Тема 16. Символьні рядки

Рядки як масиви символів. Зберігання та ініціалізація рядків. Основні функції бібліотеки string.h.

Тема 17. Структури, об'єднання, перелічення

Оголошення структур та об'єднань. Створення змінних відповідних типів. Доступ до елементів. Перелічення. Використання складених структур даних.

Тема 18. Робота з файлами у мові C

Поняття потоку. Текстові й бінарні файли. Відкриття та закриття файлів. Стандартні функції читання й запису.

Розділ 7. Рекурсія та зв'язані структури даних

Тема 19. Рекурсія

Рекурсивні функції. Пряма та опосередкована рекурсія. Базовий випадок і рекурсивний крок. Приклади застосування рекурсії.

Тема 20. Зв'язані списки

Структури прямого та послідовного доступу. Однонаправлені зв'язані списки: формування, вставлення та видалення вузлів. Рекурсивна обробка списків. Двонаправлені списки.

Тема 21. Деревя

Основні поняття, пов'язані з деревами. Формування двійкового дерева. Обходи дерев. Двійкові дерева пошуку.

Розділ 8. Абстрактні типи даних

Тема 22. Лінійні абстрактні типи даних

Поняття абстрактного типу даних. Послідовність, стек і черга. Основні операції та способи реалізації. Множина, хешування, відображення і словники.

Тема 23. Ієрархічний абстрактний тип даних «дерево»

Операції над деревами. Способи реалізації деревоподібних структур.

Тема 24. Мережевий абстрактний тип даних «граф»

Основні поняття теорії графів. Задачі, у яких застосовуються графи. Способи представлення графів за допомогою різних структур даних.

Розділ 9. Сортування

Тема 25. Прості методи внутрішнього сортування

Постановка задачі сортування. Внутрішнє та зовнішнє сортування. Оцінювання складності. Сортування вибором, включеннями та обміном.

Тема 26. Удосконалені методи внутрішнього сортування

Турнірне сортування. Пірамідальне сортування. Швидке сортування.

Тема 27. Методи зовнішнього сортування

Злиття впорядкованих послідовностей. Метод сортування фон Неймана. Сортування природним злиттям.

3. Структура навчальної дисципліни

Розподіл годин наведено для денної форми здобуття освіти. Заочна (дистанційна) форма освітньо-професійною програмою не передбачена.

1 семестр

Назви розділів і тем	Усього	Лекції	Практичні	Лабораторні	Інд.	Самостійна робота
<i>Розділ 1. Початкові відомості про алгоритми і програми. Мова програмування Python</i>						
Тема 1. Алгоритми та їх основні структури	19	6	3	0	0	10
Тема 2. Типи даних	20	6	3	0	0	11
Тема 3. Процедурне програмування	20	5	3	0	0	12
Тема 4. Тестування, налагодження, винятки. Робота з файлами	19	5	2	0	0	12
Тема 5. Аналіз алгоритмів	20	5	2	0	0	13
Тема 6. Основи об'єктно-орієнтованого програмування в Python	22	5	3	0	0	14
Усього за семестр	120	32	16	0	0	72

2 семестр

Назви розділів і тем	Усього	Лекції	Практичні	Лабораторні	Інд.	Самостійна робота
<i>Розділ 2. Загальні поняття мови програмування C</i>						
Тема 7. Базові елементи мови C	13	3	2	0	0	8
Тема 8. Типи даних мови C	13	3	2	0	0	8
<i>Розділ 3. Операції, оператори та масиви</i>						
Тема 9. Операції та їх пріоритети	12	2	2	0	0	8
Тема 10. Оператори мови C	16	3	3	0	0	10
Тема 11. Масиви	16	3	3	0	0	10
<i>Розділ 4. Функції у мові C</i>						
Тема 12. Призначення, означення і використання функцій	16	3	3	0	0	10
Тема 13. Область видимості та класи зберігання	13	2	2	0	0	9
<i>Розділ 5. Вказівники та динамічні змінні</i>						
Тема 14. Поняття вказівника	16	3	3	0	0	10
Тема 15. Вказівники і масиви	19	3	4	0	0	12
<i>Розділ 6. Складені типи даних та робота з файлами</i>						
Тема 16. Символьні рядки	15	2	3	0	0	10
Тема 17. Структури, об'єднання, перелічення	17	3	3	0	0	11
Тема 18. Робота з файлами у мові C	14	2	2	0	0	10
Усього за семестр	180	32	32	0	0	116

3 семестр

Назви розділів і тем	Усього	Лекції	Практичні	Лабораторні	Інд.	Самостійна робота
<i>Розділ 7. Рекурсія та зв'язані структури даних</i>						
Тема 19. Рекурсія	16	0	2	0	0	14
Тема 20. Зв'язані списки	16	0	2	0	0	14
Тема 21. Дерева	17	0	2	0	0	15

Назви розділів і тем	Усього	Лекції	Практичні	Лабораторні	Інд.	Самостійна робота
<i>Розділ 8. Абстрактні типи даних</i>						
Тема 22. Лінійні абстрактні типи даних	16	0	2	0	0	14
Тема 23. Ієрархічний абстрактний тип даних «дерево»	16	0	2	0	0	14
Тема 24. Мережевий абстрактний тип даних «граф»	17	0	2	0	0	15
<i>Розділ 9. Сортування</i>						
Тема 25. Прості методи внутрішнього сортування	18	0	2	0	0	16
Тема 26. Удосконалені методи внутрішнього сортування	17	0	1	0	0	16
Тема 27. Методи зовнішнього сортування	17	0	1	0	0	16
Усього за семестр	150	0	16	0	0	134

4. Теми практичних занять

1 семестр

№ з/п	Назва теми	Кількість годин
1	Алгоритми та їх основні структури: блок-схеми, псевдокод, реалізація базових алгоритмів у Python.	3
2	Типи даних Python: числа, рядки, списки, словники, множини; операції та перетворення.	3
3	Процедурне програмування: функції, параметри, модулі, елементи рекурсії.	3
4	Тестування, налагодження, винятки та робота з текстовими, CSV і JSON-файлами.	2
5	Аналіз алгоритмів: коректність, складність, реалізація алгоритмів пошуку та сортування.	2
6	Основи ООП у Python: класи, атрибути, методи, наслідування, поліморфізм.	3
	Разом	16

2 семестр

№ з/п	Назва теми	Кількість годин
1	Структура програми мовою C. Компіляція, базовий синтаксис, стандартні бібліотеки.	2
2	Типи даних C. Введення та виведення даних.	2
3	Операції, пріоритети та побітові операції.	2
4	Умовні оператори, switch, цикли, break і continue.	3
5	Одновимірні та багатовимірні масиви. Операції над матрицями.	3
6	Функції: прототипи, параметри, аргументи та структуризація програм.	3
7	Області видимості, класи зберігання та багатофайлові проєкти.	2
8	Вказівники: адресація, розіменування, використання у функціях.	3
9	Вказівники і масиви. Динамічне виділення пам'яті та динамічні масиви.	4
10	Символьні рядки та функції бібліотеки string.h.	3
11	Структури, об'єднання та перелічення.	3
12	Текстові й бінарні файли; стандартні операції читання та запису.	2

№ з/п	Назва теми	Кількість годин
	Разом	32

3 семестр

№ з/п	Назва теми	Кількість годин
1	Рекурсивні функції та типові рекурсивні алгоритми.	2
2	Однонаправлені та двонаправлені зв'язані списки.	2
3	Двійкові дерева та двійкові дерева пошуку.	2
4	Реалізація стеку, черги, множини та словника.	2
5	Операції над деревами та способи реалізації ієрархічних структур.	2
6	Представлення графів і розв'язання базових задач на графах.	2
7	Сортування вибором, включеннями та обміном.	2
8	Пірамідалне та швидке сортування.	1
9	Алгоритми злиття та зовнішнього сортування.	1
	Разом	16

5. Завдання для самостійної роботи

1 семестр

№ з/п	Види, зміст самостійної роботи	Кількість годин
1	Опрацювання та розв'язування задач з теми 1 «Алгоритми та їх основні структури».	10
2	Опрацювання та розв'язування задач з теми 2 «Типи даних».	11
3	Опрацювання теми 3 «Процедурне програмування», вправи з функціями, модулями та рекурсією.	12
4	Самостійне тестування, налагодження програм і робота з файлами за темою 4.	12
5	Аналіз коректності та складності алгоритмів; підготовка і виконання розрахунково-графічної роботи № 1.	13
6	Опрацювання принципів ООП і виконання програмних завдань за темою 6.	14
	Разом	72

2 семестр

№ з/п	Види, зміст самостійної роботи	Кількість годин
1	Базові елементи мови C, структура програми, компіляція та стандартні бібліотеки.	8
2	Типи даних C та введення/виведення.	8
3	Операції, пріоритети, побітові операції.	8
4	Оператори керування потоком; підготовка і виконання розрахунково-графічної роботи № 1.	10
5	Масиви та матриці; підготовка і виконання розрахунково-графічної роботи № 2.	10

№ з/п	Види, зміст самостійної роботи	Кількість годин
6	Функції у мові С.	10
7	Області видимості, класи зберігання та багатофайлові проекти.	9
8	Вказівники та їх використання у функціях.	10
9	Вказівники, динамічна пам'ять і динамічні масиви; підготовка і виконання розрахунково-графічної роботи № 3.	12
10	Символьні рядки.	10
11	Структури, об'єднання та перелічення.	11
12	Робота з текстовими й бінарними файлами.	10
	Разом	116

3 семестр

№ з/п	Види, зміст самостійної роботи	Кількість годин
1	Опрацювання рекурсивних алгоритмів і програмування рекурсивних функцій.	14
2	Зв'язані списки; підготовка і виконання розрахунково-графічної роботи № 1.	14
3	Дерева та дерева пошуку; підготовка і виконання розрахунково-графічної роботи № 2.	15
4	Лінійні абстрактні типи даних: стек, черга, множина, словник.	14
5	Ієрархічні структури даних і операції над деревами.	14
6	Графи та способи їх представлення.	15
7	Прості методи внутрішнього сортування; підготовка розрахунково-графічної роботи № 3.	16
8	Удосконалені методи внутрішнього сортування.	16
9	Методи зовнішнього сортування; завершення розрахунково-графічної роботи № 3.	16
	Разом	134

6. Індивідуальні завдання

Навчальним планом передбачено розрахунково-графічні роботи, які виконуються в межах самостійної роботи здобувачів освіти.

1 семестр

Розрахунково-графічна робота № 1. Розробити програму мовою Python, яка зчитує набір даних із текстового або CSV-файлу, реалізує алгоритм пошуку та алгоритм сортування, виводить результати і дає можливість порівняти час виконання обраних алгоритмів.

2 семестр

Розрахунково-графічна робота № 1. Використання розгалужень і циклів мови С для обробки цілочисельних даних та обчислення скінченних сум.

Розрахунково-графічна робота № 2. Обробка одновимірних і двовимірних масивів, виконання операцій з елементами матриць, їх рядками та стовпцями.

Розрахунково-графічна робота № 3. Використання функцій, вказівників і динамічних масивів для розв'язання прикладної задачі з організацією введення/виведення даних.

3 семестр

Розрахунково-графічна робота № 1. Реалізація та дослідження операцій над однонапрямленими або двонапрямленими зв'язаними списками.

Розрахунково-графічна робота № 2. Побудова двійкового дерева або двійкового дерева пошуку, реалізація операцій обходу, пошуку, вставлення та видалення елементів.

Розрахунково-графічна робота № 3. Реалізація, тестування та порівняльний аналіз щонайменше двох алгоритмів сортування за часом виконання та асимптотичною складністю.

7. Методи навчання

Під час викладання дисципліни застосовуються компетентнісний, студентоцентризований і проблемно-орієнтований підходи, а також поєднання пояснювально-ілюстративних, практичних, проблемних, частково-пошукових і дослідницьких методів навчання.

- лекції з демонстрацією алгоритмів, фрагментів програмного коду та прикладів розв'язання задач;
- практичні заняття з індивідуальним і груповим розв'язанням алгоритмічних та програмних задач;
- аналіз, тестування, налагодження й рефакторинг програмного коду;
- самостійне опрацювання навчальних матеріалів, документації мов програмування та рекомендованих джерел;
- виконання розрахунково-графічних робіт і захист отриманих програмних результатів;
- консультації, обговорення альтернативних алгоритмічних рішень та взаємний аналіз коду;
- використання LMS Moodle, засобів відеоконференцій та інших цифрових інструментів для підтримки очного і дистанційного компонентів навчання.

8. Методи контролю

- поточне усне та письмове опитування за матеріалами тем;
- перевірка виконання практичних і самостійних програмних завдань;
- демонстрація та пояснення здобувачем власного програмного коду;
- контрольна робота у 1 семестрі;
- перевірка і захист розрахунково-графічних робіт: 1 робота у 1 семестрі, 3 роботи у 2 семестрі та 3 роботи у 3 семестрі;
- підсумковий семестровий контроль: залік у 1 семестрі; екзамен у 2 та 3 семестрах.

9. Схема нарахування балів

Оцінювання здійснюється за 100-бальною системою. У кожному семестрі до 60 балів здобувач освіти може отримати за поточний контроль, самостійну роботу та індивідуальні завдання, до 40 балів - за підсумковий семестровий контроль. Для допуску до підсумкового контролю здобувач освіти повинен набрати не менше 30 балів за поточну роботу.

1 семестр

T1	T2	T3	T4	T5	T6	Контрольна робота	РГР №1	Залікова робота	Сума
5	5	5	5	5	5	15	15	40	100

T1-T6 - теми 1-6. Поточна робота за темами - 30 балів; контрольна робота - 15 балів; розрахунково-графічна робота - 15 балів; залікова робота - 40 балів.

2 семестр

T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	РГР 1	РГР 2	РГР 3	Екзамен	Сума
3	3	3	3	3	3	3	3	3	3	3	3	8	8	8	40	100

T7-T18 - теми 7-18. Поточна робота за темами - 36 балів; три розрахунково-графічні роботи - по 8 балів (24 бали); екзамен - 40 балів.

3 семестр

T19	T20	T21	T22	T23	T24	T25	T26	T27	РГР 1	РГР 2	РГР 3	Екзамен	Сума
4	4	4	4	4	4	4	4	4	8	8	8	40	100

T19-T27 - теми 19-27. Поточна робота за темами - 36 балів; три розрахунково-графічні роботи - по 8 балів (24 бали); екзамен - 40 балів.

Критерії оцінювання навчальних досягнень

Поточні тематичні завдання оцінюються з урахуванням правильності алгоритму, коректності програмної реалізації, якості тестування, здатності пояснити код і обґрунтувати обраний спосіб розв'язання. Кількість балів за кожну тему визначається схемою відповідного семестру.

Частка від максимального бала	Критерії
90-100% від максимальної кількості балів	Завдання виконано повністю й коректно; алгоритм обґрунтовано; код працездатний, структурований і протестований; здобувач самостійно пояснює рішення та відповідає на додаткові запитання.
70-89%	Основну частину завдання виконано правильно; наявні окремі неточності або незначні помилки, які не змінюють загальної логіки рішення та можуть бути виправлені після короткого уточнення.
50-69%	Продемонстровано базове розуміння теми; завдання виконано частково або містить суттєві помилки; здобувач потребує допомоги для виправлення алгоритму чи програмної реалізації.
0-49%	Завдання не виконано або виконано фрагментарно; алгоритмічне рішення відсутнє чи некоректне; здобувач не може пояснити основні принципи розв'язання.

Критерії оцінювання контрольної роботи (1 семестр, максимум 15 балів)

13-15 балів - повне і правильне виконання завдань з обґрунтуванням алгоритмів; 10-12 балів - правильний загальний хід розв'язання за наявності незначних помилок; 8-9 балів - базові методи застосовано правильно, але

частина завдань виконана неповно; 1-7 балів - фрагментарне виконання з істотними помилками; 0 балів - роботу не виконано.

Критерії оцінювання розрахунково-графічних робіт

Для роботи в 1 семестрі максимальна оцінка становить 15 балів, для кожної роботи у 2 та 3 семестрах - 8 балів. Оцінюються: відповідність завданню, коректність алгоритму, працездатність і якість коду, повнота тестування, аналіз отриманих результатів, самостійність виконання та здатність захистити роботу. Оцінювання здійснюється пропорційно до наведених вище рівнів 90-100%, 70-89%, 50-69% та 0-49% від максимальної кількості балів.

Критерії оцінювання залікової / екзаменаційної роботи (максимум 40 балів)

Бали	Критерії
36-40	Теоретичний зміст засвоєно повністю; програмні та алгоритмічні завдання виконано коректно; відповіді аргументовані, рішення самостійні.
28-35	Матеріал засвоєно в основному; більшість завдань виконано правильно, можливі окремі неточності або помилки.
20-27	Матеріал засвоєно частково; базові поняття та методи відтворюються, але в практичних завданнях є суттєві помилки.
0-19	Теоретичний зміст не засвоєно на достатньому рівні; практичні навички не сформовані; завдання не виконано або виконано переважно некоректно.

Використання технологій штучного інтелекту

Під час вивчення навчальної дисципліни використання технологій штучного інтелекту (ШІ) здійснюється відповідно до Політики використання технологій штучного інтелекту Харківського національного університету імені В. Н. Каразіна.

У межах дисципліни застосовуються такі моделі використання ШІ:

1. Модель «ШІ заборонено».

Використання технологій штучного інтелекту не допускається під час виконання контрольних заходів, спрямованих на перевірку індивідуальних знань і сформованості компетентностей здобувача освіти, зокрема: усного опитування, контрольної роботи, заліку, екзамену, якщо інше не визначено викладачем, та інших видів оцінювання, для яких викладачем встановлено обмеження щодо використання ШІ.

2. Модель «ШІ як асистент».

Під час підготовки до практичних занять, виконання самостійних і розрахунково-графічних робіт дозволяється використання ШІ для пошуку та систематизації інформації, пояснення складних понять, аналізу прикладів, генерації ідей, перевірки синтаксису, пошуку можливих помилок, удосконалення структури коду та оформлення пояснювальних матеріалів. Фінальні алгоритмічні рішення, програмний код, висновки, аналіз та інтерпретація результатів повинні бути критично перевірені й опрацьовані здобувачем освіти.

3. Модель «ШІ як інструмент».

Для окремих практичних, проєктних або дослідницьких завдань використання технологій штучного інтелекту може бути складовою виконання роботи. У таких випадках здобувачі освіти можуть використовувати генеративні моделі ШІ для генерації, аналізу й оптимізації програмного коду, пояснення алгоритмів, підготовки тестових даних, порівняння альтернативних програмних рішень і проведення експериментів. Усі результати, отримані за допомогою ШІ, підлягають критичному аналізу, перевірці та доопрацюванню здобувачем освіти.

Декларування використання ШІ.

У разі використання технологій штучного інтелекту здобувач освіти зобов'язаний зазначити назву сервісу або моделі ШІ, дату використання та характер допомоги (генерація ідей, пошук інформації, аналіз даних, генерація або аналіз коду, редагування тексту тощо). За вимогою викладача здобувач освіти повинен надати використані запити (prompts) або пояснити спосіб використання інструментів ШІ. Недеклароване використання технологій штучного інтелекту або подання згенерованого ШІ матеріалу як власного результату навчальної діяльності розглядається як порушення принципів академічної доброчесності відповідно до внутрішніх нормативних документів Університету.

Визнання результатів неформальної освіти

Визнання результатів навчання, отриманих у неформальній та/або інформальній освіті, здійснюється відповідно до чинних нормативних документів Харківського національного університету імені В. Н. Каразіна. Автоматичне зарахування балів за конкретні зовнішні курси цією робочою програмою не встановлюється; можливість і обсяг зарахування визначаються за результатами співвіднесення підтверджених результатів навчання зі змістом та результатами навчання дисципліни.

Шкала оцінювання

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка для чотирирівневої шкали	Оцінка для дворівневої шкали
90-100	відмінно	зараховано
70-89	добре	зараховано
50-69	задовільно	зараховано
0-49	незадовільно	не зараховано

10. Рекомендована література

Основна література

1. Matthes, E. Python Crash Course. 3rd ed. No Starch Press, 2023.
2. Ramalho, L. Fluent Python. 2nd ed. O'Reilly Media, 2022.
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. Introduction to Algorithms. 4th ed. MIT Press, 2022.
4. Seacord, R. C. Effective C: An Introduction to Professional C Programming. No Starch Press, 2020.

5. King, K. N. C Programming: A Modern Approach. 2nd ed. W. W. Norton & Company, 2008.

Допоміжна література

1. Skiena, S. S. The Algorithm Design Manual. 3rd ed. Springer, 2020.
2. Sweigart, A. Beyond the Basic Stuff with Python: Best Practices for Writing Clean Code. No Starch Press, 2020.
3. Kernighan, B. W., Ritchie, D. M. The C Programming Language. 2nd ed. Prentice Hall, 1988.
4. Miller, B. N., Ranum, D. L. Problem Solving with Algorithms and Data Structures Using Python. Franklin, Beedle & Associates, 2011.

11. Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1. Python 3 Documentation. The Python Tutorial.
<https://docs.python.org/3/tutorial/>
2. Python 3 Documentation. <https://docs.python.org/3/>
3. cppreference.com. C language reference. <https://en.cppreference.com/w/c>
4. MIT OpenCourseWare. Introduction to Computer Science and Programming in Python. <https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/>
5. Runestone Academy. Problem Solving with Algorithms and Data Structures using Python. <https://runestone.academy/>
6. Harvard University. CS50: Introduction to Computer Science.
<https://cs50.harvard.edu/x/>