

**КЛАСИЧНИЙ ПРИВАТНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ**

**СИЛАБУС**

**навчальної дисципліни**  
**«АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ»**

**КОНТАКТНА ІНФОРМАЦІЯ ТА ТЕХНІЧНОЇ ДОПОМОГА**  
(включаючи електронну пошту, робочий час / місцезнаходження тощо).

<b>Викладач (-і)</b>	Ковтун Володимир Андрійович
<b>Контактний тел.</b>	+38(096) 829-53-69 (внутр. 224)
<b>E-mail:</b>	kovtun.v.a.92@gmail.com
<b>Сторінка курсу на сайті підтримки навчальних програм КПУ</b>	<a href="http://www.zhu.edu.ua/cpu_edu/course/view.php?id=5257">http://www.zhu.edu.ua/cpu_edu/course/view.php?id=5257</a>
<b>Консультації</b>	<i>Консультації off-line:</i> за графіком консультацій викладача інформаційно-комунікаційними технологіями: ZOOM - <a href="https://us02web.zoom.us/j/7844225252?pwd=Q1A1bkFPMExwXmQvWHBhVDJjODJvZz09">https://us02web.zoom.us/j/7844225252?pwd=Q1A1bkFPMExwXmQvWHBhVDJjODJvZz09</a> Google Meet: <a href="https://meet.google.com/qgq-ncrp-yha">https://meet.google.com/qgq-ncrp-yha</a>

**АНОТАЦІЯ**

Навчальна дисципліна «Алгоритми та структури даних» є нормативною для студентів першого (бакалаврського) рівня вищої освіти галузі знань 01 Освіта/Педагогіка за спеціальністю 014 Середня освіта, освітня програма: Інформатика. Згідно з навчальним планом денної форми навчання вивчення дисципліни заплановано на 4 семестр (2 курс).

Особливістю курсу для спеціальності 014 Середня освіта освітня програма є спрямованість на формування у майбутніх учителів навичок розробки «гарних алгоритмів» та їхнього застосування як центрального поняття програмування, що дозволяє використовувати алгоритмічний аналіз.

Курс передбачає вивчення теоретичного фундаменту обчислювальних наук, властивостей та класифікації алгоритмів, способів їх графічного вираження, а також опанування мови Python, зокрема її типів даних, операторів, керуючих конструкцій та будованих структур.

Освітній процес з дисципліни здійснюється за такими формами: навчальні заняття; самостійна робота; контрольні заходи. Видами навчальних занять згідно з навчальним планом є лекції; практичні заняття, а також консультації.

Практичні заняття передбачають побудову блок-схем згідно з вимогами стандарту, роботу в середовищі розробки IDLE, реалізацію логічних розгалужень та циклів, а також практичне застосування базових алгоритмів пошуку та сортування даних.

Самостійна робота проводиться під час аудиторних занять та в час, вільний від обов'язкових навчальних занять, без участі викладача шляхом самостійного опрацювання лекційного матеріалу, виконання індивідуальних завдань з кожного модуля курсу. Повний курс лекційного матеріалу та методичні рекомендації до виконання індивідуальних домашніх завдань розміщено на сторінках дисципліни сайту підтримки навчальних програм університету.

Консультації призначені для роз'яснення студентам теоретичних або практичних питань.

Засвоєння навчального матеріалу перевіряється за допомогою поточного контролю, який здійснюється на практичних заняттях у формі усних відповідей, самостійних робіт та перевірки виконання завдань, виконання практичних робіт. Для визначення результатів модульного та підсумкового контролю використовується система накопичення балів, яка стимулює систематичну роботу студента протягом семестру.

Підсумковий (семестровий) контроль після завершення 4 семестру здійснюється у формі екзамену.

## ФОРМАТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**Загальна кількість годин – 120 год., у т. ч. 64 годин аудиторних занять і 56 годин самостійної роботи студента. Кількість кредитів ECTS – 4.**

Всього кредитів	Всього годин	Аудиторних годин	У тому числі			Сам. робота
			Лекц.	Лабор.	Семін. (практ.)	
4	120	64	16	16	32	56

## ОЗНАКИ ДИСЦИПЛІНИ

Курс (рік навчання)	Семестр	Цикл підготовки	Нормативна/ вибіркова
2	4	професійна	нормативна

## МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### Мета навчальної дисципліни

Метою навчальної дисципліни є формування у студентів теоретичного фундаменту сучасних обчислювальних наук через вивчення основ алгоритмізації та опанування мови програмування Python як інструменту для створення ефективних програмних рішень. Навчання спрямоване на те, щоб студенти навчилися формально описувати обчислювальні процедури, розробляти власні алгоритми та візуалізувати їх за допомогою блок-схем, що є ключовим етапом перед безпосереднім написанням коду. Програма курсу передбачає глибоке вивчення базових і розширених структур даних, опанування керуючих конструкцій, а також дослідження класичних алгоритмів пошуку та сортування з урахуванням їхньої швидкодії та трудомісткості. У підсумку студенти мають оволодіти методиками раціонального вибору алгоритмів та розробки програм для автоматизації процесів обробки інформації в різних галузях знань.

### Завдання навчальної дисципліни

- формування теоретичного фундаменту сучасних обчислювальних наук шляхом вивчення теорії алгоритмів та математичної логіки;
- опанування формалізації поняття «алгоритм» та вивчення різних алгоритмічних систем для опису процесів переробки інформації;
- вивчення засобів автоматизації, що забезпечуються наявністю відповідних алгоритмів

для розв'язання обчислювальних задач;

- дослідження властивостей алгоритмів та критеріїв оцінки їхньої якості;
- ознайомлення з принципами побудови та аналізу базових структур даних та класичних алгоритмів пошуку й сортування;
- навчання раціональному вибору алгоритмів для вирішення конкретних прикладних завдань з урахуванням обмежень на час та пам'ять.

## **ЗАПЛАНОВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ**

### **У результаті вивчення дисципліни студенти повинні знати:**

- основні визначення та властивості алгоритму, такі як детермінованість, дискретність, зрозумілість та результативність;
- історію розвитку теорії алгоритмів, включаючи внесок Евкліда, Аль-Хорезмі, Т'юринга, Черча та Поста;
- форми запису алгоритмів;
- базові керуючі конструкції мови Python;
- вбудовані структури даних Python;
- алгоритми пошуку даних;
- алгоритми сортування;
- параметри оцінки складності алгоритмів, зокрема асимптотичний час виконання та використання додаткової пам'яті.

### **Після вивчення дисципліни студенти повинні вміти:**

- формально описувати обчислювальні процедури, перетворюючи вхідні аргументи на вихідні результати;
- розробляти графічні схеми алгоритмів згідно з вимогами стандартів для візуалізації логічної структури задачі;
- складати план розв'язання задачі, визначаючи вхідні, проміжні та вихідні величини;
- програмувати мовою Python, використовуючи середовище розробки IDLE для написання та налагодження коду;
- застосовувати різні типи даних та здійснювати їхню конвертацію;
- реалізовувати складні розгалуження та ітераційні процеси для обробки великих масивів даних;
- ефективно маніпулювати колекціями та виконувати операції над множинами;
- виконувати асимптотичний аналіз складності алгоритмів та здійснювати раціональний вибір методу для досягнення максимальної швидкодії.

Відповідно до освітньо-професійної програми підготовки бакалавра галузі знань 01 Освіта/Педагогіка за спеціальністю 014 Середня освіта, освітня програма: Інформатика вивчення дисципліни «Алгоритми та структури даних» сприяє формуванню **компетентностей та програмних результатів навчання:**

### **Інтегральна компетентність:**

Здатність розв'язувати складні спеціалізовані задачі у галузі середньої освіти, що передбачає застосування теоретичних знань і практичних умінь з наук предметної спеціальності, педагогіки, психології, теорії та методики навчання і характеризується комплексністю та невизначеністю умов організації освітнього процесу в закладах середньої освіти.

### **Загальні компетентності:**

ЗК 2. Здатність до абстрактного мислення, аналізу та синтезу. Володіння методами логічного та системного аналізу, здатність виявляти закономірності в інформаційних процесах та розв'язувати складні завдання предметної області.

ЗК 3. Інформаційно-комунікаційна компетентність. Здатність ефективно

використовувати сучасні ІКТ для пошуку, обробки, критичної оцінки та поширення інформації в освітньому та соціальному просторах.

### Спеціальні (фахові) компетенції:

СК 1. Теоретико-методологічна предметна компетентність. Глибоке розуміння фундаментальних основ інформатики (алгоритмізація, дискретна математика, логіка) та здатність інтерпретувати їх для учнів різних вікових категорій.

СК 3. Програмно-технологічна компетентність. Здатність розробляти програмні продукти, використовуючи сучасні мови програмування, знання архітектури ПК, операційних систем та мереж.

### Програмні результати навчання:

РН 1. Знає та розуміє фундаментальні засади математики та теоретичної інформатики, достатні для формування предметних компетентностей з інформатики.

РН 3. Використовує сучасні мови, методи та технології програмування для розв'язання практичних завдань і створення програмних продуктів.

## ПЛАН КУРСУ

Назва змістових модулів та тем	Лекц.	Лабор.	Практ (сем.)	Завдання для самостійної роботи
<b>Змістовий модуль № 1. Основи алгоритмізації та вступ до Python</b>				
Тема №1. Вступ до алгоритмів. Властивості та види.	2	2	4	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Теоретичний та практичний аспекти застосування теорії алгоритмів у сучасній науці. 2) Еквівалентність формальних визначень алгоритму у трактуваннях Кнута, Колмогорова та Маркова. 3. Виконання індивідуального завдання № 1. 4. Виконання тестового завдання на сайті підтримки навчальних програм.
Тема №2. Способи подання алгоритмів.	2	2	4	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Правила словесно-формульного запису алгоритмів із застосуванням математичних обчислень. 2) Методика перевірки правильності складеної блок-схеми на конкретних значеннях аргументів. 3. Виконання індивідуального завдання № 2. 4. Виконання тестового завдання на сайті підтримки навчальних програм.
Тема №3. Основи Python: Змінні, типи даних та оператори.	2	2	4	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Відмінності у принципах роботи компіляторів та інтерпретаторів.

Назва змістових модулів та тем	Лекц.	Лабор.	Практ (сем.)	Завдання для самостійної роботи
				2) Використання функції type() для динамічного визначення типу даних змінної. 3. Виконання індивідуального завдання № 3. 4. Виконання тестового завдання на сайті підтримки навчальних програм.
Тема №4. Керуючі конструкції Python: Умовні оператори та цикли.	2	2	4	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Представлення логічних значень у цифровій формі як 1 та 0. 2) Вимоги до синтаксичного оформлення коду за допомогою відступів для позначення вкладених дій. 3. Виконання індивідуального завдання № 4. 4. Виконання тестового завдання на сайті підтримки навчальних програм. 5. Підготовка до модульного контролю за темами 1-4.
<b>Змістовий модуль № 2. Структури даних та пошукові/сортувальні алгоритми</b>				
Тема №5. Базові вбудовані структури даних Python: Списки та Кортежі.	2	2	4	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Синтаксичні правила створення порожніх колекцій та кортежів з одного елемента. 2) Реалізація вкладених структур даних на прикладі списків у списках (матриць). 3. Виконання індивідуального завдання № 5. 4. Виконання тестового завдання на сайті підтримки навчальних програм.
Тема №6. Розширені вбудовані структури даних Python: Множини та Словники.	2	2	4	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Перевірка дочірніх та батьківських зв'язків між множинами через issubset() та issuperset(). 2) Створення словників на основі вкладених списків за допомогою функції dict(). 3. Виконання індивідуального завдання № 6. 4. Виконання тестового завдання на сайті підтримки навчальних програм.
Тема №7. Алгоритми пошуку даних.	2	2	4	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань:

Назва змістових модулів та тем	Лекц.	Лабор.	Практ (сем.)	Завдання для самостійної роботи
				1) Порівняльна ефективність лінійного та двійкового пошуку в різних умовах. 2) Реалізація двійкового пошуку у вигляді рекурсивної функції. 3. Виконання індивідуального завдання № 7. 4. Виконання тестового завдання на сайті підтримки навчальних програм.
Тема №8. Алгоритми сортування.	2	2	4	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Використання непрямого сортування для впорядкування великих записів за допомогою вказівників. 2) Вплив вибору опорного елемента та рандомізації на швидкодію алгоритму Quicksort. 3. Виконання індивідуального завдання № 8. 4. Виконання тестового завдання на сайті підтримки навчальних програм. 5. Підготовка до модульного контролю за темами 5-8.

## ФОРМИ КОНТРОЛЮ ТА КРИТЕРІЇ ОЦІНЮВАННЯ

У процесі вивчення навчальної дисципліни «Алгоритми та структури даних» використовуються наступні види контролю:

1. Поточний контроль – здійснюється протягом семестру шляхом перевірки виконання тестових завдань, виконання лабораторних робіт, модульних контрольних робіт тощо. За змістом він включає перевірку ступеню засвоєння студентом навчального матеріалу, який охоплюється темою лекційного заняття, уміння самостійно опрацьовувати навчально-методичну літературу, здатність осмислювати зміст теми, уміння публічно та письмово представити певний матеріал, а також виконання завдань самостійної роботи.

2. Підсумковий семестровий контроль – здійснюється у формі письмового екзамену відповідно до графіка освітнього процесу шляхом визначення ступеню засвоєння студентом навчальної дисципліни за результатами виконання обов'язкових завдання поточного (модульного) контролю.

Для оцінювання студентів використовується система накопичування балів. Згідно з «Положенням про організацію освітнього процесу в КПУ» підсумкова оцінка з дисципліни виставляється за 100-бальною шкалою з наступним переведенням у національну шкалу та шкалу ECTS.

Бали нараховуються за виконання завдань аудиторної роботи, практичних, лабораторних робіт, контрольних (модульних) завдань, тестів.

Результати поточного контролю здобувачів вищої освіти є складовими елементами підсумкової оцінки з дисципліни.

Оцінка рівня роботи студента протягом семестру під час навчальних занять та самостійної роботи здійснюється у межах 80 балів. Вага екзамену у підсумковій оцінці складає 20 балів.

## РОЗПОДІЛ БАЛІВ ПОТОЧНОГО ТА ПІДСУМКОВОГО (СЕМЕСТРОВОГО) ОЦІНЮВАННЯ

Поточне оцінювання під час навчальних занять та самостійної роботи								Екзамен	Сума
Змістовий модуль 1				Змістовий модуль 2					
T1	T2	T3	T4	T5	T6	T7	T8	20	100
10	10	10	10	10	10	10	10		

### Критерії оцінювання

Для оцінювання відповідей студентів з навчальної дисципліни «Алгоритми та структури даних» використовуються наступні **критерії**:

– рівню «відмінно» відповідає теоретично правильна і вичерпна відповідь на поставлене питання, у якій студент показав всебічне системне знання програмного матеріалу; засвоєння основної та додаткової літератури; чітке володіння понятійним апаратом, методами, методиками та інструментами, вивчення яких передбачене програмою дисципліни; уміння використовувати знання для аналізу життя економічного суспільства та аргументувати своє ставлення до відповідних категорій, закономірностей, випадковостей, суспільних явищ і процесів;

– рівню «добре» відповідає теоретично правильна, але не вичерпна відповідь на поставлене запитання, в цілому повне знання програмного матеріалу, успішне виконання запропонованого завдання і засвоєння матеріалу основної літератури;

– рівню «задовільно» відповідає у цілому правильна відповідь на поставлене питання, в якій студент показав достатній рівень знань з основного програмного матеріалу дисципліни, але не зміг переконливо аргументувати свою відповідь, помилився у використанні понятійного апарату, показав недостатні знання рекомендованої літератури;

– рівню «незадовільно» відповідає неправильна або неповна відповідь на запитання, у якій студент продемонстрував значні прогалини у знаннях з основного програмного матеріалу; ухилився від аргументувань; показав незадовільні знання понятійного апарату і спеціальної літератури чи взагалі нічого не відповів.

### Шкала оцінювання: 100-бальна, національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
75-81	C		
67-74	D		
60-66	E	задовільно	не зараховано з можливістю повторного складання
35-59	FX	незадовільно з можливістю повторного складання	
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### Підручники та навчальні посібники

1. Кормен Т., Лейзерсон Ч., Рівест Р., Стайн К. Вступ до алгоритмів (4-е видання). – К.: К.І.С., 2023. – 1300 с.
2. Коноваленко І. В. Алгоритмізація та програмування мовою Python: Навчальний посібник. – Тернопіль: ТНТУ, 2021. – 192 с.
3. Трофименко О. Г., Прокоп Ю. В. Алгоритмізація та програмування: Підручник. – Одеса: Фенікс, 2020. – 260 с.
4. Бхаргава А. Грокаємо алгоритми. Ілюстрований посібник для програмістів та цікавих. – Харків: Ранок, 2019. – 288 с.

### Статті у наукових фахових виданнях

1. Мельник О. В. Методичні особливості навчання алгоритмізації в курсі інформатики. – К.: Науковий часопис НПУ імені М.П. Драгоманова, 2022. – Вип. 15. – С. 45-52.
2. Сидоренко В. М. Аналіз ефективності бінарного пошуку в упорядкованих структурах даних. – Львів: Вісник ЛНУ, 2021. – № 12. – С. 110-118.
3. Петренко І. С. Сортування Хоара: практичні аспекти реалізації мовою Python. – Дніпро: Системи та технології, 2023. – № 2. – С. 34-40.
4. Коваль А. М. Використання словників та множин у Python для обробки великих масивів даних. – Харків: Радіoeлектроніка та інформатика, 2020. – № 4. – С. 67-73.

### Автореферати дисертацій

1. Шевченко О. Л. Алгоритми та структури даних у задачах інтелектуального аналізу: автореф. дис. канд. техн. наук. – К., 2021. – 20 с.
2. Бондаренко С. В. Моделі та методи автоматизації обробки даних у керуючих системах: автореф. дис. канд. техн. наук. – Харків, 2022. – 18 с.
3. Іванов Д. О. Оптимізація алгоритмів внутрішнього сортування для розподілених обчислень: автореф. дис. канд. техн. наук. – Львів, 2023. – 22 с.
4. Гриценко Ю. П. Розробка ефективних алгоритмів пошуку у неструктурованих базах даних: автореф. дис. канд. техн. наук. – Одеса, 2019. – 21 с.

## ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

### Українські освітні ресурси

1. Prometheus: [prometheus.org.ua](https://prometheus.org.ua) – Провідна платформа масових відкритих онлайн-курсів в Україні, зокрема з основ програмування та Python.
2. Освіта.ua: [osvita.ua](https://osvita.ua) – Інформаційний ресурс з актуальними матеріалами для викладачів інформатики та студентів педагогічних ВНЗ.
3. На Урок: [naurok.com.ua](https://naurok.com.ua) – Освітній проект для вчителів, що містить розробки уроків та лабораторних робіт з алгоритмізації.
4. Python.ua: [python.com.ua](https://python.com.ua) – Спільнота українських розробників, новини та навчальні матеріали з мови програмування Python.

### Програмне забезпечення та документація

1. Python Official Documentation: [docs.python.org](https://docs.python.org) – Офіційний сайт з вичерпною документацією, описом вбудованих функцій та модулів мови.
2. IDLE (Python): [python.org/downloads](https://python.org/downloads) – Інтегроване середовище розробки та навчання для Python, що розглядається в курсі.
3. Visual Studio Code: [code.visualstudio.com](https://code.visualstudio.com) – Потужний редактор коду з підтримкою Python, зручний для виконання складних лабораторних робіт.
4. Draw.io: [app.diagrams.net](https://app.diagrams.net) – Онлайн-інструмент для створення блок-схем та графічного подання алгоритмів згідно зі стандартами

### **Електронні підручники та освітні платформи**

1. W3Schools Python Tutorial: [w3schools.com/python](http://w3schools.com/python) – Популярний інтерактивний ресурс для вивчення Python з великою кількістю практичних прикладів.
2. Coursera: [coursera.org](http://coursera.org) – Міжнародна платформа з курсами від провідних університетів світу з алгоритмів та структур даних.
3. Stepik: [stepik.org](http://stepik.org) – Освітня платформа з інтерактивними завданнями та автоматичною перевіркою коду Python.
4. EdX: [edx.org](http://edx.org) – Ресурс з академічними курсами з Computer Science та основ алгоритмізації від MIT та Harvard.

### **Фахові спільноти та ресурси**

1. Stack Overflow: [stackoverflow.com](http://stackoverflow.com) – Найбільша у світі спільнота програмістів для пошуку відповідей на складні технічні питання.
2. DOU: [dou.ua](http://dou.ua) – Найбільша IT-спільнота України з форумами, статтями та оглядами сучасних технологій програмування.
3. GitHub: [github.com](http://github.com) – Платформа для хостингу коду та спільної розробки проектів, джерело прикладів реалізації класичних алгоритмів.
4. Kaggle: [kaggle.com](http://kaggle.com) – Спільнота для фахівців з Data Science, де Python використовується як основний інструмент аналізу даних.