

КЛАСИЧНИЙ ПРИВАТНИЙ УНІВЕРСИТЕТ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

СИЛАБУС

навчальної дисципліни
«ОСНОВИ ПРОГРАМУВАННЯ»

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

Викладач (-і)	Ковтун Володимир Андрійович
Контактний тел.	+38(096) 829-53-69 (внутр. 224)
E-mail:	kovtun.v.a.92@gmail.com
Сторінка курсу на сайті підтримки навчальних програм КПУ	http://www.zhu.edu.ua/cpu_edu/course/view.php?id=4979
Консультації	<i>Консультації off-line:</i> за графіком консультацій викладача інформаційно-комунікаційними технологіями: ZOOM - https://us02web.zoom.us/j/7844225252?pwd=Q1A1bkFPMExwxdmQvWHBhVDJjODJvZz09 Google Meet: https://meet.google.com/qgq-ncrp-yha

АНОТАЦІЯ

Навчальна дисципліна «Основи програмування» 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 1 семестр (1 курс).

Особливістю курсу для спеціальності 014 Середня освіта освітня програма є орієнтація на формування не лише навичок кодування, а й розуміння методики навчання програмуванню, що дозволить майбутнім вчителям ефективно пояснювати складні концепції учням, використовуючи наочність та простоту синтаксису Python, який ідеально підходить для початкового вивчення програмування у школі.

Курс передбачає вивчення повного циклу розробки програмного забезпечення від постановки задачі до візуалізації результатів, охоплюючи як процедурний підхід (функції, модулі), так і роботу з різними типами даних, що забезпечує базу для подальшого вивчення об'єктно-орієнтованого програмування та аналізу даних.

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

Лабораторні заняття передбачають виконання індивідуальних практичних завдань за варіантами, що включають написання програмного коду, його тестування, налагодження в

середовищі IDLE, роботу з бібліотеками та оформлення звітів з аналізом отриманих результатів.

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 1 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	48	-	56

ОЗНАКИ ДИСЦИПЛІНИ

Курс (рік навчання)	Семестр	Цикл підготовки	Нормативна/вибіркова
1	1	професійна	нормативна

МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мета навчальної дисципліни

Метою навчальної дисципліни є формування у студентів системи теоретичних знань та практичних навичок у галузі алгоритмізації та програмування мовою Python, розвиток алгоритмічного мислення, здатності декомпонувати складні задачі на підзадачі та реалізовувати їх за допомогою сучасних інструментальних засобів розробки для подальшого використання у професійній педагогічній діяльності.

Завдання навчальної дисципліни

- ознайомлення з синтаксисом та основними конструкціями мови Python;
- вивчення базових структур даних та алгоритмів їх обробки;
- формування навичок роботи з інтегрованим середовищем розробки (IDLE);
- набуття вмінь використовувати стандартні бібліотеки Python для вирішення математичних та прикладних задач.

ЗАПЛАНОВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

У результаті вивчення дисципліни студенти повинні знати:

- основні поняття та історію розвитку мови Python;
- типи даних, оператори та правила запису виразів;
- синтаксис керуючих конструкцій (розгалуження, цикли);
- принципи структурного програмування та роботи з функціями;
- методи роботи з файловою системою та візуалізації даних.

Після вивчення дисципліни студенти повинні вміти:

- встановлювати та налаштовувати середовище програмування;
- розробляти та налагоджувати програмний код мовою Python;
- використовувати функції та модулі для структурування коду;
- здійснювати операції введення-виведення даних, в тому числі з файлів;
- будувати графіки функцій та діаграми для аналізу даних.

Відповідно до освітньо-професійної програми підготовки бакалавра галузі знань 01 Освіта/Педагогіка за спеціальністю 014 Середня освіта, освітня програма: Інформатика вивчення дисципліни «Основи програмування» 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 загальної 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.

ЗК 7. Здатність до креативності та інновацій. Здатність генерувати нові ідеї, виявляти гнучкість у професійних ситуаціях та застосовувати творчий підхід до розв'язання практичних проблем.

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
Змістовий модуль № 1. Основи алгоритмізації та обробка текстових даних				
Тема №1. Основні поняття мови Python.	3	8	-	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Коментарі в коді та стиль оформлення програм. 2) Помилки інтерпретації та їх виправлення. 3. Виконання індивідуального завдання № 1. 4. Виконання тестового завдання на сайті підтримки навчальних програм.
Тема №2. Логічні оператори та цикли.	3	8	-	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Вкладені цикли та їх застосування. 2) Генерація випадкових чисел за допомогою модуля random. 3. Виконання індивідуального завдання № 2. 4. Виконання тестового завдання на сайті підтримки навчальних програм.
Тема №3. Робота з рядками у мові Python.	2	8	-	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Екрановані послідовності в рядках. 2) Генератори словників. 3. Виконання індивідуального завдання № 3. 4. Виконання тестового завдання на сайті підтримки навчальних програм. 5. Підготовка до модульного контролю за темами 1-3.
Змістовий модуль № 2. Структуризація коду, збереження даних та візуалізація				
Тема №4. Функції в Python.	3	8	-	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Використання функцій як аргументів інших функцій. 2) Конструкція <code>if __name__ == "__main__":</code> . 3. Виконання індивідуального завдання № 4. 4. Виконання тестового завдання на сайті підтримки навчальних програм.
Тема №5. Робота з файлами в Python.	3	8	-	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання

Назва змістових модулів та тем	Лекц.	Лабор.	Практ (сем.)	Завдання для самостійної роботи
				теоретичних питань: 1) Порівняння файлів за допомогою модуля filecmp. 2) Копіювання та перейменування файлів. 3. Виконання індивідуального завдання № 5. 4. Виконання тестового завдання на сайті підтримки навчальних програм.
Тема №6. Побудова графіків математичних функцій у мові Python.	2	8	-	1. Опрацювання лекційного матеріалу, 2. Самостійне опрацювання теоретичних питань: 1) Додавання текстових приміток на графік. 2) Використання логарифмічного масштабу осей. 3. Виконання індивідуального завдання № 6. 4. Виконання тестового завдання на сайті підтримки навчальних програм. 5. Підготовка до модульного контролю за темами 4-6.
Разом	16	48		56

ФОРМИ КОНТРОЛЮ ТА КРИТЕРІЇ ОЦІНЮВАННЯ

У процесі вивчення навчальної дисципліни «Основи програмування» 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 опрацювати 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 60 балів. Вага екзамену у підсумковій оцінці складає 40 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	40	100
10	10	10	10	10	10		

Критерії оцінювання

Оцінка ECTS	Бали	Характеристика виконання завдань (програмування)
A (Відмінно)	90–100	Студент вільно використовує функції, модулі та складні структури даних. Продемонстровано здатність до декомпозиції задач та візуалізації даних. Програмний код написаний без помилок, відповідає стандартам оформлення.
B (добре)	82–89	Студент впевнено володіє синтаксисом та базовими алгоритмами. Код працює коректно, але можливі незначні недоліки в стилі оформлення або коментарях.
C (Добре)	75–81	Студент успішно виконує індивідуальні завдання, вміє працювати з файлами (читання/запис) та стандартними бібліотеками. Програма виконує основні функції, проте студент не завжди може аргументувати вибір конкретного алгоритму чи методу обробки даних.
D (Задовільно)	67–74	Студент знає синтаксис керуючих конструкцій (if, for, while). Вміє створювати прості скрипти, але припускається логічних помилок у складних алгоритмах. Програма працює, але є помилки в логіці побудови циклів чи розгалужень. Слабке володіння інструментами налагодження в IDLE.
E (Задовільно)	60–66	Студент знає основні конструкції мови (типи даних, оператори), виконує завдання за зразком але має значні труднощі при створенні власних функцій чи роботі з файлами.
FX (Незадовільно)	35–59	Студент демонструє значні прогалини у знаннях програмного матеріалу. не розуміє різниці між типами даних або не вміє підключити модуль. Написаний код містить критичні помилки, які не дозволяють програмі запуснитися (помилки інтерпретації). Потрібне повторне складання.
F (Незадовільно)	0–34	Студент не володіє базовою термінологією та не здатний написати навіть найпростіший алгоритм. Повна відсутність розуміння принципів алгоритмізації. Потрібне повторне навчання.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Підручники та навчальні посібники

1. Ковалюк Т. В. Основи програмування. Python: навч. посіб. – Київ: Видавничий дім «Кондор», 2022. – 380 с.
2. Трофименко О. Г. Алгоритмізація та програмування мовою Python: навч. посіб. / О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко. – Одеса: Фенікс, 2020. – 248 с.
3. Бондаренко М. А. Python для початківців: теоретичні основи та практичні завдання: підручник. – Харків: ХНУРЕ, 2021. – 290 с.
4. Федорчук А. В. Python: від основ до розробки прикладних програм. – Львів: Видавництво Львівської політехніки, 2019. – 412 с.

Статті у наукових фахових виданнях

1. Іванов І. І. Методика навчання основ програмування мовою Python у закладах середньої освіти // Інформаційні технології в освіті. – 2023. – № 2. – С. 45-52.
2. Петренко О. В. Використання бібліотеки Matplotlib для візуалізації математичних моделей у навчальному процесі // Комп'ютерно-інтегровані технології. – 2021. – Вип. 45. – С. 112-118.
3. Сидоренко М. П. Порівняльний аналіз ефективності вивчення Python та C++ як першої мови програмування // Педагогічні науки: теорія та практика. – 2020. – № 3. – С. 89-94.
4. Коваленко Т. М. Застосування проектного методу при вивченні модульного програмування на Python // Науковий вісник Ужгородського університету. Серія: Педагогіка. – 2022. – Вип. 1. – С. 67-73.

Автореферати дисертацій

1. Мельник А. С. Розвиток алгоритмічного мислення старшокласників засобами мови програмування Python: автореф. дис. ... канд. пед. наук: 13.00.02. – Київ, 2021. – 20 с.
2. Ткаченко В. І. Методична система навчання об'єктно-орієнтованого програмування майбутніх вчителів інформатики: автореф. дис. ... канд. пед. наук: 13.00.02. – Вінниця, 2019. – 20 с.
3. Гончарук О. П. Формування компетентності з візуалізації даних у студентів технічних спеціальностей: автореф. дис. ... канд. пед. наук: 13.00.10. – Харків, 2022. – 21 с.
4. Демченко С. В. Інтеграція хмарних технологій у процес навчання програмування мовою Python: автореф. дис. ... канд. пед. наук: 13.00.02. – Полтава, 2020. – 19 с.

ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

Українські освітні ресурси

1. Prometheus: prometheus.org.ua – платформа масових відкритих онлайн-курсів, що містить курси з основ програмування на Python українською мовою.
2. ITVDN: itvdn.com/ua – освітній онлайн-ресурс для IT-спеціалістів, який пропонує відеоуроки та інтерактивні завдання з Python.
3. Python-Україна: python.org.ua (спільнота) – ресурс для обміну досвідом українських розробників, новини та події зі світу Python.
4. Електронна бібліотека НАПН України: lib.iitta.gov.ua – репозитарій наукових та навчально-методичних матеріалів, де розміщено посібники з інформатики та програмування.

Програмне забезпечення та документація

1. Офіційний сайт Python: python.org – джерело для завантаження інтерпретатора, доступу до офіційної документації (Python Docs) та стандартних бібліотек.
2. JetBrains PyCharm Edu: jetbrains.com/pycharm-edu – інтегроване середовище розробки з освітніми можливостями для вивчення Python.
3. Matplotlib: matplotlib.org – офіційна документація бібліотеки для побудови графіків,

містить приклади та tutoriали.

4. Python Package Index (PyPI): pypi.org – репозитарій програмного забезпечення для мови програмування Python.

Електронні підручники та освітні платформи

1. Coursera: coursera.org – міжнародна платформа з курсами від провідних університетів (наприклад, Python for Everybody).

2. W3Schools Python: w3schools.com/python – популярний ресурс з простими та зрозумілими прикладами коду, інтерактивним редактором.

3. Real Python: realpython.com – ресурс з якісними tutoriалами та статтями для поглибленого вивчення аспектів мови.

4. Codecademy: codecademy.com/learn/learn-python-3 – інтерактивна платформа для вивчення синтаксису Python на практиці.

Фахові спільноти та ресурси

1. Stack Overflow: stackoverflow.com – найбільша спільнота розробників для отримання відповідей на технічні питання щодо Python.

2. GitHub: github.com – платформа для хостингу коду, де можна знайти приклади відкритих проектів на Python.

3. DOU.ua: dou.ua – найбільша спільнота програмістів України, статті про розвиток ринку праці та технологій.

4. Reddit Python: reddit.com/r/Python – новинний агрегатор та форум для обговорення новин та проектів у світі Python.