

Шифр та назва спеціальності	G7 (174) Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Назва освітньої програми	Комп'ютерні технології та програмування в автоматизованих системах керування
Назва дисципліни	Прикладне програмування на Python
Вид (основна, вибіркова)	Вибіркова
Блок дисципліни (залишити блок, що відповідає дисципліні)	1. Алгоритмізація та програмування (зокрема, різні парадигми: структурне, ООП)
Кількість студентів (поточний рік)	22
Курс/Семестр	3 курс, осінній семестр (5 семестр)
Викладач	Караман Дмитро Григорович, старший викладач кафедри «Автоматика та управління в технічних системах» (АУТС), Національний технічний університет «Харківський політехнічний інститут» (НТУ «ХПІ») https://web.kpi.kharkov.ua/auts/ua/

ЗАГАЛЬНА ІНФОРМАЦІЯ ПРО ДИСЦИПЛІНУ

Анотація	Дисципліна "Прикладне програмування на Python" орієнтована на студентів, які вже мають базові знання з алгоритмів, структур даних і основ програмування іншими мовами. Метою курсу є ознайомлення з синтаксисом, парадигмами та інструментальними засобами мови Python, а також формування навичок розробки прикладних програмних рішень у сфері автоматизації, комп'ютерно-інтегрованих технологій та робототехніки. Особливу увагу приділено створенню програм для систем автоматизованого керування, вбудованих пристроїв, обробки й аналізу даних, розробці застосунків з графічним інтерфейсом, а також програмуванню обчислювальних процесів у реальному часі. У рамках навчального процесу студенти опановують підходи до створення Python-програм для взаємодії з різними технологіями, протоколами зв'язку, а також засоби тестування, налагодження та швидкої розробки прототипів функціональних програмних модулів сучасних інженерних систем.								
Мета навчальної дисципліни	Формування у студентів здатності використовувати мову програмування Python для розробки прикладних програмних рішень, що відповідають завданням галузі автоматизації, комп'ютерно-інтегрованих технологій та робототехніки.								
Типи занять та контролю	Лекції, лабораторні роботи, самостійна робота. Підсумковий контроль – диференційований залік. Усі навчальні матеріали надаються в електронному вигляді, що дозволяє проведення занять як у аудиторній, так і у дистанційній формі. Навчальні матеріали адаптовано до проведення занять з використанням LMS Microsoft Teams або Google Classroom. Проводиться проміжний модульний контроль у вигляді письмових контрольних робіт або електронного тестування.								
Загальний обсяг (кредитів)	4	Лекції (занять)	16	Лабораторні (занять)	8	Практичні (занять)	–	Самостійна (годин)	72
Попередні дисципліни	Програмування, Алгоритми та структури даних								

МАТЕРІАЛЬНО-ТЕХНІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ

Студент повинен мати базові навички та знання користувача ПК, вмінні здійснювати інформаційний пошук матеріалів за заданою темою, бажано володіння англійською мовою. Для виконання лабораторних робіт та завдань самостійної роботи необхідно використовувати персональний комп'ютер з ОС Windows або Linux (від 2 ГБ ОЗП, 1 ГБ дискового простору, доступ в Інтернет) з встановленим інтерпретатором мови Python версії не нижче 3.10 а також додатковими засобами автоматизованої розробки (редакторами програмного коду) за бажанням студента (наприклад, Visual Studio Code). Бажаним є користування засобами контролю версій програмного забезпечення (Git), а також наявність онлайн-репозиторію для збереження початкового коду виконаних завдань та самостійних робіт (GitHub)

СТРУКТУРА ТА ЗМІСТ ДИСЦИПЛІНИ

№	Теоретична складова <i>Назва, перелік питань або анотація лекції</i>	Годин	Практична складова <i>Опис та приклад завдання, а також посилання на методичні матеріали</i>	Годин	Інструменти, засоби та технології
Тема 1 – Базові відомості про мову Python					
1	Роль програміста у сучасному світі, передумови входження в професію, критерії вибору мови програмування. Базові відомості про мову Python, сильні та слабкі сторони. Встановлення, налаштування, необхідні умови та вимоги, базові та додаткові інструменти, способи виконання програм на мові Python.	2	Опитування засвоєних знань за допомогою короткого тестування (виконується після лекції, або за рахунок часу, виділеного на самостійну роботу)	1	Тести в MS Forms або Google Forms
2	Основи синтаксису, особливості та обмеження мови, стандартні вбудовані типи (об'єкти типів), правила опису літералів та ідентифікаторів, основні оператори та вирази, розуміння змінних у Python (об'єктна природа даних), вбудовані функції у Python, зокрема, перетворення типів, вводу та виводу даних.	2	Опитування засвоєних знань за допомогою короткого тестування (виконується після лекції, або за рахунок часу, виділеного на самостійну роботу)	1	Тести в MS Forms або Google Forms
3	Рядкові дані - способи задавання (кодування), збереження та оперування (оператори, які визначені над рядковими даними), індексація та утворення фрагментів, функції, пов'язані з рядковими даними, методи об'єктів рядкових літералів, схожість та відмінність рядкових і байтових даних.	2	Опитування засвоєних знань за допомогою короткого тестування (виконується після лекції, або за рахунок часу, виділеного на самостійну роботу)	1	Тести в MS Forms або Google Forms
4	Списки (list) та кортежі (tuple), Множини (set) та словники (dictionary). Об'єктна природа та методи. Способи створення, доступ до елементів, фрагментація. Додавання, зміна та видалення елементів. Очищення та видалення самого об'єкту колекції. Ітеративний характер колекцій. Сортування, створення копії, об'єднання та розділення.	2	Опитування засвоєних знань за допомогою короткого тестування (виконується після лекції, або за рахунок часу, виділеного на самостійну роботу)	1	Тести в MS Forms або Google Forms

<p>5 Конструкції керування процесом виконання коду: умовне розгалуження та цикли. Умовне розгалуження за допомогою конструкції if..elif..else. Розгалуження за допомогою конструкції match..case. Організація циклів за допомогою for та while. Функція range() та обхід ітеративних об'єктів. Ключові слова break, continue, pass. Блок else в конструкціях for та while.</p>	<p>2 Лабораторна робота. Засвоєння основ мови програмування Python. Створення програми, яка виконує приймання даних, введених користувачем, виконує їх обробку та виводить результат відповідно до індивідуального завдання. В залежності від завдання приймається рішення використовувати ти чи інші можливості мови Python</p>	<p>2 Інтерпретатор Python, редактор коду (VS Code), система контролю версій коду git, акаунт на GitHub</p>
---	---	---

Тема 2 – Структурування програмного коду на Python

<p>6 Функції. Оголошення та використання. Способи передавання аргументів у функцію. Повернення результатів. Рекурсія. Зони видимості змінних: локальні та глобальні. Вбудовані функції. Лямбда-функції. Функції-генератори (yield).</p>	<p>2 Опитування засвоєних знань за допомогою короткого тестування (виконується після лекції, або за рахунок часу, виділеного на самостійну роботу)</p>	<p>1 Тести в MS Forms або Google Forms</p>
<p>7 Модулі та пакети модулів. Підключення та використання інших модулів у складі мови Python. Способи підключення. Вбудовані стандартні модулі. Створення та використання власного модулю. Додавання та використання сторонніх модулів та пакетів (які не входять до складу стандартних модулів Python). Система реєстрації та поширення пакетів PyPI.</p>	<p>2 Опитування засвоєних знань за допомогою короткого тестування (виконується після лекції, або за рахунок часу, виділеного на самостійну роботу)</p>	<p>1 Тести в MS Forms або Google Forms</p>
<p>8 Створення стійкого та безпечного коду. перехоплення та обробка виключних випадків у програмах (exception handling), вбудовані типи помилок та виключних ситуацій. Конструкція try .. except (+else +finally +raise).</p>	<p>2 Лабораторна робота. Створення функціонально завершеного застосунку на мові Python. Створюється застосунок, який виконує ускладнене алгоритмічне завдання відповідно до індивідуального завдання (статистична обробка, сортування, багатетапні розрахунки, тощо). При створенні застосунку вимагається використання функцій та об'єднання їх у користувацький модуль, а також застосування методів створення стійкого та безпечного коду.</p>	<p>2 Інтерпретатор Python, редактор коду (VS Code), система контролю версій коду git, акаунт на GitHub</p>

Тема 3 – Створення прикладних програм на Python

<p>9 Робота з файлами в Python. Текстові файли. Читання та запис даних. Розташування файлів, робочий каталог та робота з файловою системою ПК. Робота з файлами у бінарному форматі. Структурування даних у файлах, спеціальні формати файлів: csv, xml, json. Спеціальні пакети по роботі з файлами цих форматів, читання, парсинг, запис даних та структур даних. Опрацювання параметрів командного рядка.</p>	<p>2 Лабораторна робота. Робота з файлами користувацьких даних. Вивчення методів та конструкцій по роботі з файлами у мові Python. Виконання індивідуального завдання, яке потребує завантаження початкових даних з файлу та збереження результатів роботи у файл.</p>	<p>2 Інтерпретатор Python, редактор коду (VS Code), система контролю версій коду git, акаунт на GitHub</p>
---	---	---

10 Об'єктно-орієнтоване програмування в Python. Загальне розуміння про об'єктно-орієнтоване програмування та способи подання ОПП та його принципів у мові Python. Знайомство з об'єктами у Python на прикладі вбудованих класів. Створення, опис та використання об'єктів у мові Python. Створення та використання власного класу. Принципи побудови об'єктно-орієнтованого ПЗ.	2	Опитування засвоєних знань за допомогою короткого тестування (виконується після лекції, або за рахунок часу, виділеного на самостійну роботу)	1	Тести в MS Forms або Google Forms
11 Створення програм з графічним інтерфейсом користувача для програм на Python. Принципи та методи побудови ПЗ з графічним інтерфейсом. Огляд різних бібліотек для створення графічного інтерфейсу у Python (Tkinter, PyQt, Kivy). Пакет Tk (Tkinter). Основні віджети та контейнери. Налаштування функціоналу та вигляду елементів інтерфейсу.	2	Лабораторна робота. Створення програми з графічним інтерфейсом. Покращення зручності взаємодії з користувачем через використання засобів графічного інтерфейсу. Засвоєння вбудованих можливостей створення програм з графічним інтерфейсом у Python.	2	Інтерпретатор Python, редактор коду (VS Code), система контролю версій коду git, акаунт на GitHub

Тема 4 – Мережеве програмування на Python

12 Створення програм на Python з використанням мережевих інтерфейсів. Поняття сокетів та взаємодії різних програм з їх використанням. Бібліотека sockets. Формування та обробка запитів. Клієнт-серверна технологія. Принцип REST. Протоколи: HTTP, HTTPS. URL-адресування. Методи запитів HTTP (GET, POST). Коди станів у відповідях. Вбудовані бібліотеки, що працюють з HTTP: urllib, http.	2	Лабораторна робота. Створення програм, які взаємодіють через мережевий інтерфейс. Вивчення та засвоєння принципів взаємодії програм через механізм сокетів. Практичне використання ізольованих застосунків, які взаємодіють через сокети.	2	Інтерпретатор Python, редактор коду (VS Code), система контролю версій коду git, акаунт на GitHub
13 Веб-програмування на Python. Огляд варіантів фреймворків та їх можливостей: Django, Flask та ін. Огляд можливостей фреймворку Flask. Встановлення та налаштування. Створення базового додатку (вебсайту). Маршрути у Flask. Робота з динамічними URL-адресами. Використання шаблонів сторінок. Робота з формами та отримання даних від користувача. Робота з джерелами даних (SQLAlchemy). Деплоймент веб-додатку.	2	Лабораторна робота. Створення веб-застосунку. Використання веб-технологій для створення прикладного ПЗ. Можливості мови Python для створення та розгортання веб-застосунків, їх практичне застосування.	2	Інтерпретатор Python, редактор коду (VS Code), система контролю версій коду git, акаунт на GitHub

Тема 5 – Інші способи прикладного використання мови Python

14	Багатопоточне програмування у Python. Визначення задач, які можливо розділяти на паралельні потоки. Способи побудови багатопоточного ПЗ. Основні принципи роботи з потоками. Створення та управління виконанням потоків. Бібліотека threading та її використання для багатопоточного виконання програм у Python. Особливості та обмеження мови Python по роботі з потоками.	2	Опитування засвоєних знань за допомогою короткого тестування (виконується після лекції, або за рахунок часу, виділеного на самостійну роботу)	1	Тести в MS Forms або Google Forms
15	Використання Python для роботи з базами даних. Створення, наповнення, підтримка файлової бази даних SQLite. Способи підключення до серверних БД. Робота з базами даних типу No-SQL. Статистична обробка даних (Data Science) та методи обробки великих об'ємів даних (Big Data) за допомогою Python. Пакети numpy, scipy та matplotlib. (Використання Jupyter Notebook та Google Colab).	2	Лабораторна робота. Обробка та візуалізація даних. Вивчення можливостей мови Python для математичного та статистичного опрацювання великих об'ємів даних. Практичний приклад обробки даних та наочної візуалізації результатів у вигляді графіків та діаграм.	2	Інтерпретатор Python, редактор коду (VS Code), система контролю версій коду git, акаунт на GitHub
16	Використання Python для програмування вбудованих систем (embedded systems). Способи створення та запуску програм на мові Python на вбудованих пристроях. Платформа Raspberry Pi. Спеціальні реалізації мови Python: MicroPython та CircuitPython для роботи на мікроконтролерних платформах для вбудованих систем (ESP8266, ESP32, Raspberry Pi Pico)	2	Лабораторна робота. Створення програм для вбудованих систем. Використання мови Python для розробки та виконання програм на базі вбудованих систем та мікроконтролерних пристроїв. Практичний приклад розробки вбудованого ПЗ.	2	Інтерпретатор Python, редактор коду (VS Code), система контролю версій коду git, акаунт на GitHub

ТЕМИ ТА ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

№	Назва та опис завдання	Методи контролю та критерії оцінювання	Годин
1	Системи контролю початкового коду програмних продуктів. Зберігання, резервування та контроль версій. Основні поняття та терміни. Приклади практичного застосування. Система Git та ресурс GitHub	Тестування, демонстрація створеного репозиторію на GitHub	2
2	Основи архітектури програмного забезпечення. Різновиди архітектур: монолітна, багат шарова, багаторівнева, сервіс-орієнтована. Залежність архітектури від призначення ПЗ, середовища виконання, структури команди розробки та способу менеджменту.	Тестування	2
3	Сучасні методології проектування програмного забезпечення. Різновиди методологій: каскадна ("водоспад"), інкрементальна, ітеративна, гнучка (agile), спіральна та ін. Залежність методології від інших аспектів розробки.	Тестування	2

4	Документування початкового коду ПЗ. Стандарти та загально прийняті практики документування. Системи автоматичного складання документації за початковим кодом.	Тестування, додаткові бали за використання засвоєних знань при виконанні лабораторних робіт	2
5	Створення безпечного коду та функціональна перевірка розробленого ПЗ. Способи попередження помилок та конструкції для проектування безпечного коду. Методи тестування та можливості мови Python для організації перевірки створеного ПЗ.	Тестування, додаткові бали за використання засвоєних знань при виконанні лабораторних робіт	2
6	Принципи роботи мережевих та інтернет-месенджерів. Створення боту для месенджера Telegram.	Тестування, виконання індивідуального завдання зі створення боту	2
7	Принципи організації та роботи багатопоточних програм. Визначення задач, які можливо розділяти на паралельні потоки.	Тестування	2
8	Вступ до програмованої електроніки та вбудованих систем. Особливості розробки ПЗ для вбудованих систем. Організація та робота технологій зв'язку в IoT: робота систем за підпискою (брокери MQTT)	Тестування	2

ПРОЄКТ (за наявністю)

№	Назва та опис завдання	Метод контролю та захисту	Строки виконання
1	(навчальним планом не передбачено)		

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ ТА НАВЧАЛЬНІ МАТЕРІАЛИ

Основні

№	Назва	До теми (вказати номер)
1	Програмування в Python. Теорія і практика / Олексій Васильєв. – Вид-во: Ліра-К, 2023. – 462 с. ISBN : 9786175205136	1-3
2	Head First Python. Легкий для сприйняття довідник / Пол Беррі. – Вид-во: Фабула, 2021. – 624 с. ISBN : 9786175220191	1-3
3	Пришвидшений курс Python. Практичний, проектно-орієнтований вступ до програмування / Ерік Маттес. – Видавництво Старого Лева, 2021. – 600 с. ISBN : 9786176798538	1-3
4	Путівник мовою програмування Python. Вивчення основ програмування для початківців: електронний підручник / Олександр Мізюк (gtmizyuk@gmail.com) 03.12.2020. URL: https://pythonguide.rozh2sch.org.ua/	1-4
5	Підручник з Python. Офіційна документація Python. URL: https://docs.python.org/uk/3/tutorial/index.html	1-5
6	Початок роботи з Python і робота з даними: лабораторний практикум з навчальної дисципліни «Основи програмування Python (дисципліна вибору 02)» для студентів спеціальностей 122 «Комп'ютерні науки» та 126 «Інформаційні системи та технології». Харків: НТУ «ХПІ», 2022. 97 с. https://repository.kpi.kharkov.ua/handle/KhPI-Press/57807	5
7	Уроки Python для початківців / Програмування на Пітон з нуля // Youtube-канал Школа програмування (плейліст) https://www.youtube.com/playlist?list=PL7vq4D0vOpQb-nious5dTzUF9GEMp6jil	1-4
8	Real Python: навчальні матеріали з програмування мовою Python [Електронний ресурс, англ. мовою]. — URL: https://realpython.com/	1-5

Додаткові

№	Назва	До теми (вказати номер)
1	Об'єктно-орієнтоване програмування мовою PYTHON : Конспект лекцій [Електронний ресурс] : навч. посіб. для студ. спеціальності 153 «Мікро- та наносистемна техніка» освітньої програми «Мікро- та наноелектроніка» / КПІ ім. Ігоря Сікорського ; уклад.: Д. Д. Татарчук, Ю. В. Діденко. – Електронні текстові данні (1 файл: 1,3 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 129 с.	1-3
2	McKinney, W. (2022). Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter (3rd ed.). O'Reilly Media. Available at: https://wesmckinney.com/book/	5
3	Структурне програмування : методичні вказівки до виконання індивідуальних завдань з курсу «Технологія програмування» для студентів всіх рівнів та форм навчання спеціальностей 113 «Прикладна математика» та 122 «Комп'ютерні науки» / уклад. О. О. Ларін, М. І. Шаповалова. — Харків : НТУ «ХПІ». 2020. — 48 с. https://repository.kpi.kharkov.ua/handle/KhPI-Press/49164	Для СР
4	Алгоритми і структури даних: практикум: навч. посіб. / Н. К. Стратієнко, М. Д. Годлевський, І. О. Бородіна. – Харків: НТУ «ХПІ», 2017. – 224 с. ISBN 978-617-05-0247-6 URL: https://repository.kpi.kharkov.ua/handle/KhPI-Press/50239	1-3
5	Shovic J. C. Raspberry Pi IoT Projects: Prototyping Experiments for Makers. 2-е вид. – Berkeley, CA: Apress, 2021. – 278 с. – ISBN 978-1-4842-6911-4.	5
6	Halfacree G., Everard B. Get Started with MicroPython on Raspberry Pi Pico. – 2021. – Cambridge: Raspberry Pi Press. – 137 с. – ISBN 978-1-912047-86-4. URL: https://magazine.raspberrypi.com/books/micropython-pico	5

КОНТРОЛЬНІ ЗАХОДИ

№	Назва та опис	Методи контролю та критерії оцінювання
1	Поточні контрольні заходи	Короткі тести після вивчення теоретичного матеріалу, оцінки за 100-бальною системою та шкалою ECTS
2	Перевірка лабораторних робіт	Звіти про виконання, що прикріплюються до завдань у електронній навчальній системі, або завантажуються у репозиторій Github, код створених програм на Github, оцінки за 100-бальною системою та шкалою ECTS
3	Модульні контрольні роботи	Розгорнуті тести за темами цілого змістовного модулю (тема 1 – тема 5), оцінки за 100-бальною системою та шкалою ECTS
4	Підсумковий контроль (диференційований залік)	Опитування під час особистої зустрічі, тестування, вирішення завдання перед викладачем (за наявності достатньої кількості оцінок в результаті поточного контролю може не проводитись)

РЕЗУЛЬТАТИ НАВЧАННЯ

Вивчення даної дисципліни передбачає досягнення програмних результатів навчання згідно освітньої програми, а саме: вміти застосовувати сучасні інформаційні технології та мати навички розробки комп'ютерних програм на основі синтезованих алгоритмів із використанням мов високого рівня та технологій об'єктно-орієнтованого програмування, використовувати бази даних та інтернет-ресурси, вміти проектувати багаторівневі системи керування і збору даних для формування бази параметрів процесу та їх візуалізації за допомогою засобів людино-машинного інтерфейсу, вміти створювати та використовувати прикладне програмне забезпечення для розв'язування типових інженерних задач у галузі автоматизації, зокрема, математичного моделювання, автоматизованого проектування, керування базами даних, методів комп'ютерної графіки для створення графічних інтерфейсів, тощо.

ЗВ'ЯЗОК ІЗ РИНКОМ ПРАЦІ

Спеціальність/професія, підготовці до діяльності в якій читається курс

Інше (інженер з автоматизації, інженер-програміст АСУ ТП, Embedded Systems Developer/Engineer)

Посилання на вакансії (понад 3), та їх скриншоти, в яких вказані компетенції, що формуються на курсі, як вимоги до кандидатів

<https://jobs.dou.ua/companies/soft-industry-allince/vacancies/312904/>

20 червня 2025

Senior Python/C++ Developer with hardware integration experience

📍 віддалено

We're looking for a **Senior Python/C++ Developer with strong experience in hardware integration and desktop client development** to join an ambitious R&D project in the 3D printing industry. This is an opportunity to work at the frontier of physical-digital integration – shaping how 3D content is delivered and printed in real time.

About the Project

We're building a platform where users can purchase 3D models online and instantly print them on local 3D printers with no manual setup. You will develop the client-side solution (desktop service or app) that links the cloud to a wide range of 3D printers – including plastic, metal, and concrete-based devices – through SDKs, APIs, and custom protocols.

Must-Have Requirements

- 5+ years of development experience
- Strong skills in Python and/or C++
- Experience with hardware-level communication and protocol handling
- Knowledge of G-code, STL/OBJ formats, and print workflows
- USB/Serial/WebSocket integration experience
- Cross-platform development proficiency
- Experience with CI/CD and packaging systems (MSI, DMG, DEB)
- English: Upper-Intermediate+

<https://www.work.ua/jobs/6541021/>

Вакансія від 26 червня 2025

🕒 Вище за середню



Інженер з автоматизації та комп'ютерних технологій

💰 30 000 – 130 000 грн, Уточнюється додатково

📅 6 центр рекрутингу Сил Безпілотних Систем ✓ 🇺🇦 🏆

Сили оборони; 50–250 співробітників

📍 Вся Україна

✓ Повна зайнятість. Також готові взяти студента. Досвід роботи від 1 року.



ProstoPay [Всі вакансії компанії](#)

ProstoPay – це розробник вбудованих рішень для вендингу. Наш продукт надає партнерам безперервний доступ до даних телеметрії їхнього обладнання. Якщо говорити про перспективи, то в нас кожен отримує можливості для персонального розвитку. Ви можете як вирости професійно в одному напрямку, так і змінити вектор діяльності.

[Всі вакансії](#) / [Python](#) / [Київ](#)

30 червня 2025

Python developer

📍 Київ, віддалено \$2000–3000

Ми – українська IT-компанія, що розробляє IoT-рішення для автономних бізнесів.

Що потрібно знати про нас: ProstoPay є провідним рішенням в Україні і тепер виходить у нові країни. Ми:

- розробляємо власний hard та soft
- націлені на розвиток
- є резидентами Дія.City
- працюємо в B2B

Наша сфера діяльності сповнена пригод. Наші люди – готові до викликів, націлені на розвиток, хочуть отримати досвід та досягати результатів разом із нами.

І сьогодні ми в пошуках Python-розробника, якому цікаво розвивати продукт зі складною логікою, який готовий копатися до суті навіть у тих нечастих випадках, коли код і є дока.

В ролі Python-розробника ти будеш допомагати розвивати наші продукти. Це, зокрема:

- розробка нового функціоналу в межах поточних проєктів
- підтримка та модифікація наявного функціоналу
- створення самостійних мікросервісів, таких як API та сервіси обробки даних від MQTT-брокера

Для нас важливо, щоб у стеку скілів було:


- вміння працювати з MySQL та PostgreSQL
- вміння працювати з RESTful API
- відповідальність і відкритість до дискусій та обговорень


<https://www.olx.ua/uk/obyavlenie/rabota/python-programst-IDXBu4b.html>


Опубліковано 20 червня 2025 р.

Python-програміст

 ЗАРПЛАТА
22 000 грн. / за місяць

 МІСЦЕЗНАХОДЖЕННЯ
Київ, Святошинський

 ТИП ЗАЙНЯТОСТІ
Повна зайнятість

 ГРАФІК РОБОТИ
Повний робочий день

Досвід роботи: Підходить кандидатам: Студентам

ОПИС

Обов'язки:

- Розробка та підтримка програмного забезпечення на Python.
- Участь у проєктуванні й оптимізації коду.
- Інтеграція з базами даних і зовнішніми API.
- Співпраця з командою для реалізації технічних завдань.

Вимоги:

- Досвід програмування на Python від 1 року.
- Знання основних бібліотек (наприклад, Django, Flask) і принципів ООП.
- Базове розуміння роботи з базами даних (SQL або NoSQL).
- Відповідальність і вміння працювати в команді.

Перелік компетентностей із вказаних як вимоги до вакансії, які набувають студенти, в процесі проходження дисципліни.

Здатність створювати та впроваджувати системи автоматизації для роботизованих комплексів та безпілотних платформ, використовуючи сучасні програмно-технічні засоби, навички в розробці та налаштуванні систем моніторингу, збору та обробки даних для підвищення ефективності технологічних процесів, hardware integration and desktop client development, hardware-level communication and protocol handling, Knowledge of embedded or IoT protocols, розробка та підтримка програмного забезпечення на Python, інтеграція з базами даних і зовнішніми API, досвід програмування на Python, Знання основних бібліотек (наприклад, Django, Flask) і принципів ООП, Базове розуміння роботи з базами даних (SQL або NoSQL), розвивати продукт зі складною логікою, модифікація наявного та розробка нового функціоналу знання Docker, SQLAlchemy, MQTT, FastAPI, вміння працювати з RESTful API

Інструменти оцінювання результатів навчання за дисципліною*

№	Об'єкт оцінювання (знання методів та принципів, практичні навички, командна робота тощо)	Методи контролю (тести, виконання поточних практичних завдань та їх форма: написання коду, створення діаграми Гантта, створення прототипу тощо)	Інструмент оцінювання (доступ до результатів тесту, гостьова лекція, посилання на виконані завдання, посилання на проект, присутність на захисті проектів, доступ до запису захисту тощо)
1	Теоретичні знання мови програмування Python	Тести	Посилання на результати
2	Практичні навички використання мови Python	Програмний код, звіти – матеріали, створені студентами в процесі виконання лабораторних робіт	Посилання на GitHub-репозитарії студентів, електронні звіти
3			
4			
5			
6			
7			
8			
9			
10			

***Блок. Інструменти оцінювання результатів навчання за дисципліною** — цей блок має на меті показати можливі варіанти проведення валідації за участі експертів від ІТ-компаній. Основна умова проведення валідації це попереднє погодження всіх аспектів з експертом та викладачем. Інструменти, завдяки яким це можна зробити:

- тести за темою (аналогічно технічному інтерв'ю);
- поточні завдання впродовж курсу (посилання на код на GitHub-репозитарію студента тощо);
- підсумковий проєкт (код, презентація, скринкаст презентації тощо);
- гостьові активності (гостьові лекції, відвідування контрольних активностей тощо).