

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«КАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНІКА»**



Факультет/інститут фізико-технічний

Кафедра комп'ютерної інженерії та електроніки

**СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ПРОГРАМУВАННЯ**

Освітня програма: Комп'ютерне проектування інтегральних схем

Спеціальність: G5 Електроніка, електронні комунікації,
приладобудування та радіотехніка

Галузь знань: G Інженерія, виробництво та будівництво

Затверджено на засіданні кафедри
Протокол № 13 від “26” червня 2025 р.

м. Івано-Франківськ – 2025 р.

ЗМІСТ

1. Загальна інформація	3
2. Опис дисциплін	3
3. Структура курсу	5
4. Система оцінювання курсу	10
5. Оцінювання відповідно до графіку навчального процесу	10
6. Ресурсне забезпечення	10
7. Контактна інформація	12
8. Політика навчальної дисципліни	12

1. Загальна інформація

Назва дисципліни	Програмування
Освітня програма	Комп'ютерне проектування інтегральних схем
Спеціалізація (за наявності)	-
Спеціальність	G5 Електроніка, електронні комунікації, приладобудування та радіотехніка
Галузь знань	G Інженерія, виробництво та будівництво
Освітній рівень	Бакалавр
Статус дисципліни	Очна, нормативна
Курс/семестр	2/3
Розподіл за видами занять та годинами навчання (якщо передбачені інші види, додати)	6 кредитів Лекції – 30 год. Лабораторні заняття – 30 год. Самостійна робота – 120 год.
Мова викладання	Українська
Посилання на сайт дистанційного навчання	https://d-learn.pnu.edu.ua

2. Опис дисципліни

<p>Мета та цілі курсу</p>	<p>Метою навчальної дисципліни є набуття студентами знань, професійних компетенцій, умінь і навичок програмування мовою Python та ефективного її використання для розв'язування прикладних задач. Отримані знання дозволять студенту використовувати процедурну, функційну та об'єктно-орієнтовану парадигму при розробленні обчислювальних, багатопотокових, мережевих та графічних програм.</p> <p>У результаті вивчення навчальної дисципліни студент повинен знати:</p> <ul style="list-style-type: none"> - принципи розроблення програмного забезпечення на основі процедурної, функційної та об'єктно-орієнтованої парадигм; - синтаксис, типи даних, операції, оператори, ітератори; - бібліотеки і модулі для підтримки багатопоточності і взаємодії процесів. - мережеві бібліотеки і протоколи TCP/IP та UDP/IP; - бібліотеки комп'ютерної графіки; - бібліотеки розроблення програм з графічним інтерфейсом; <p>вміти:</p> <ul style="list-style-type: none"> - вибирати інструментальні засоби та бібліотеки розроблення програмних додатків мовою Python;
---------------------------	---

	<ul style="list-style-type: none"> - працювати з інтегрованими середовищами розробки та налагодження Python програм; - розробляти програми з консольним і графічним інтерфейсом; - розробляти програми з багатьма потоками і взаємодією процесів; - розробляти мережеві програми з архітектурою клієнт сервер; <p>Дисципліна «Програмування» належить до переліку нормативних освітніх компонент освітньої програми “Комп’ютерне проектування інтегральних схем” освітнього рівня «бакалавр».</p> <p>Силабус навчальної дисципліни “Програмування” складений відповідно до освітньо-професійної програми “Комп’ютерне проектування інтегральних схем” підготовки бакалаврів спеціальності "G5 Електроніка, електронні комунікації, приладобудування та радіотехніка".</p>
Компетентності	<p>Загальні компетентності</p> <p>ЗК1. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК5. Навички використання інформаційних і комунікаційних технологій.</p> <p>ЗК8. Навички міжособистісної взаємодії.</p> <p>Спеціальні (фахові предметні) компетентності</p> <p>СК5. Здатність застосовувати відповідні математичні, наукові й технічні методи, сучасні інформаційні технології і комп’ютерне програмне забезпечення, навички роботи з комп’ютерними мережами, базами даних та Інтернет-ресурсами для вирішення інженерних задач в галузі електроніки.</p> <p>СК11. Здатність контролювати і діагностувати стан обладнання, застосовувати сучасні електронні компоненти та технічні засоби, виконувати профілактику, ремонт та технічне обслуговування електронних пристроїв та систем, монтувати, налагоджувати та ремонтувати аналогові, цифрові та оптичні модулі, розробляти та виготовляти друковані плати, розробляти програмне забезпечення для мікроконтролерів.</p>
Програмні результати навчання	<p>P5. Використовувати інформаційні та комунікаційні технології, прикладні та спеціалізовані програмні продукти для вирішення задач проектування та налагодження електронних систем, демонструвати</p>

	<p>навички програмування, аналізу та відображення результатів вимірювання та контролю.</p> <p>P12. Використовувати документацію, пов'язану з професійною діяльністю, із застосуванням сучасних технологій та засобів офісного устаткування; використовувати англійську мову, включаючи спеціальну термінологію, для спілкування з фахівцями, проведення літературного пошуку та читання текстів з технічної та фахової тематики.</p>
--	--

3. Структура курсу

№	Тема	Результати навчання	Завдання
1	<p>Вступ в Python. Область застосування і можливості Python. Інсталяція Python. Пакет рір. Віртуальне середовище. Запуск Python. Синтаксис і змінні Python.</p>	<p>Знати область застосування і можливості мови Python. Вміти інсталиувати потрібну версію Python у віртуальне середовище. Вміти запускати Python з командного рядка. Знати синтаксис Python. Вміти створювати змінні.</p>	<p>Поточний контроль, тести, завдання для самостійної роботи</p>
2	<p>Вбудовані типи даних. Операції. Числа (int, float, complex), символічні стрічки (str), булеві (bool), бінарні (bytes, bytearray, memoreview), без типу (NoneType). Перетворення типів, функції int(), float(), str(). Модифікації і зчеплення стрічок. Зрізи стрічок. Методи стрічок. Виведення і введення стрічок, функції print(), input(). Форматування стрічок. Операції Python.</p>	<p>Знати вбудовані типи даних, функції для перетворення вбудованих типів даних, методи стрічок, оператори Python (арифметичні, присвоєння, порівняння, логічні, ідентичності, членства, побітові). Вміти перетворювати вбудовані типи даних, маніпулювати стрічками, формувати стрічки, використовувати відповідні оператори.</p>	<p>Поточний контроль, тести, завдання для самостійної роботи</p>
3	<p>Вбудовані типи даних. Колекції. Кортежі, списки, словники,</p>	<p>Знати вбудовані типи даних колекції, маніпулювання елементами, організацію</p>	<p>Поточний контроль, тести,</p>

	множини - доступ, зміна і вилучення елементів, розпакування, ітерації і методи.	ітерацій, методи. Вміти об'єднувати та ітерувати колекції, використовувати методи колекцій.	завдання для самостійної роботи
4	Оператори і твердження (конструкції). Оператори арифметичні, присвоєння, порівняння, логічні, ідентичності, членства, побітові. Конструкції галуження, циклів, оброблення і збудження винятків.	Знати оператори арифметичні, присвоєння, порівняння, логічні, ідентичності, членства, побітові та конструкції (галуження, циклів, оброблення і збудження винятків). Вміти застосовувати оператори розгалуження за умовами, організувати цикли, обробляти помилки в блоках коду, створювати і обробляти власні винятки.	Поточний контроль, тести, завдання для самостійної роботи
5	Функції. Визначення функції. Передача аргументів у функцію. Рекурсивна і лямбда функції. Простори імен і область видимості. Ключові слова global, nonlocal. Властивості функції. Декоратори функцій. Анотації функцій.	Знати типи функцій, їх властивості, особливості передачі аргументів і повернення значень з функції, простори імен і область видимості. Вміти передавати аргументи у функції і повертати з них значення, створювати замкнення, фабричні функції та функції на замовлення, використовувати анотації функцій.	Поточний контроль, тести, завдання для самостійної роботи
6	Додаткові можливості процедурного програмування. Розгалуження з використанням словників. Генератори списків, словників і множин. Ітеровані об'єкти і об'єкти ітератори. Вирази генератори. Функції генератори, співпрограми. Частково підготовлені функції.	Знати реалізацію розгалуження з використанням словників, генератори списків, словників і множин, ітератори, вирази функції генератори, частково підготовлені функції, особливості функційного програмування. Вміти генерувати списки, словники множини, організувати відкладені обчислення з використанням співпрограм, застосовувати	Поточний контроль, тести, завдання для самостійної роботи

	Функційне програмування.	функційне програмування.	
7	ООП і класи. Класи і об'єкти. Методи класів <code>__init__()</code> , <code>__str__()</code> . Функція <code>super()</code> . Методи, параметр <code>self</code> . Інкапсуляція. Успадкування, атрибути і лінеаризація. Множинне успадкування. Поліморфізм.	Знати основні концепції ООП, синтаксис визначення класу, створення класу і екземплярів класу, спеціальні методи класу, простори імен і області видимості класу. Вміти створювати класи і екземпляри класів, реалізовувати концепції інкапсуляції, успадкування і поліморфізму.	Поточний контроль, тести, завдання для самостійної роботи
8	Класи. Додаткові можливості. Створення класів без атрибуту <code>__dict__</code> . Доступ до атрибутів класу і екземплярів. Дескриптори. Функтори. Декоратори класів. Абстрактні базові класи. Метакласи.	Знати атрибути класу і спеціальні методи, призначення дескрипторів і функторів, декоратори класу, призначення абстрактних базових класів та метакласів. Вміти створювати екземпляри класу і встановлювати любі їх атрибути, створювати дескриптори і функтори, розробляти декоратори класів.	Поточний контроль, тести, завдання для самостійної роботи
9	Структури даних користувача. Стек, Черга, Зв'язаний список, Геш таблиці, Дерева, Двійкові дерева, Дерева двійкового пошуку, Збалансовані двійкові дерева, Графи,	Знати структури даних, які користувач може створити на основі класів. Вміти створювати структури даних на основі класів, додавати/вилучати елементи, здійснювати пошук заданих елементів.	Поточний контроль, тести, завдання для самостійної роботи
10	Процеси. Поняття процесів. Галуження процесів, функції <code>os.fork()</code> , <code>os.exec()</code> . Взаємодія процесів. Анонімні канали. Двонаправлений обмін. Іменовані канали. Сигнали.	Знати поняття процесів, бібліотеки для створення нових процесів, механізми взаємодії між процесами, механізм асинхронних подій на основі сигналів. Вміти запускати паралельні процеси комбінацією викликів <code>fork()/exec()</code> , організувати взаємодію процесів з використанням іменованих і	Поточний контроль, тести, завдання для самостійної роботи

		неіменованих каналів, обгортати дескриптори каналів об'єктами файлів, використовувати обробники сигналів.	
11	Пакунок multiprocessing. Модуль multiprocessing, клас Process. Об'єкти Pipe, Value, Array, Queue.	Знати призначення пакунку і модуля multiprocessing, класу Process, об'єктів Pipe, Value, Array, Queue. Вміти запускати процеси з використанням методів .start(), .join() екземпляра класу Process, використовувати об'єкт Pipe для створення анонімного каналу, який з'єднує два процеси, використовувати об'єкти Value і Array для обміну даними між процесами через спільну пам'ять, використовувати чергу Queue для обміну даними між процесами.	Поточний контроль, тести, завдання для самостійної роботи
12	Потоки. Поняття потоків. Модулі thread, _thread, threading. Клас Thread. Примітиви синхронізації, блокування, керування контекстом, семафори. Модуль Queue/queue.	Знати поняття потоків, функції модуля thread, методи блокування об'єкта LockType, механізми синхронізації модуля threading, атрибути і методи класу Thread, примітиви синхронізації, механізми зв'язку між потоками на основі модуля Queue. Вміти створювати потоки і запускати в них функцію на виконання, використовувати об'єкти синхронізації потоків Lock, Condition, Event, Semaphore, Timer, Barrier.	Поточний контроль, тести, завдання для самостійної роботи
13	Мережі. Поняття архітектури клієнт-сервер. Сокет, адреса сокета, встановлення з'єднань. Модуль socket, функція socket.socket(),	Знати архітектуру клієнт/сервер, родини сокетів і родини адрес Інтернету, модуль socket, вбудовані методи об'єкта socket, модуль socketserver, можливості	Поточний контроль, тести, завдання для самостійної роботи

	об'єкт сокет і його методи. Сокет TCP/IP, UDP/IP. Створення клієнта/сервера TCP/IP, UDP/IP. Модуль socketserver. Середовище мережевого програмування Twisted.	середовища мережевого програмування twisted. Вміти створювати екземпляри сокетів, клієнт/сервер TCP/IP, клієнт/сервер UDP/IP, сервер з використанням модуля socketserver.	
14	Бібліотека Matplotlib. Побудова графіків за допомогою Pyplot. Побудова стовпчикових, кругових діаграм, гістограм.	Знати функції для роботи з графіками і діаграмами: plot(), subplot(), marker(), xlabel(), ylabel(), grid(), scatter(), bar(), hist(), pie(). Вміти створювати графіки, стовпчикові і кругові діаграми, гістограми.	Поточний контроль, тести, завдання для самостійної роботи
15	Бібліотека PyQt. Встановлення PyQt. Керування вікном застосунку. Задання координат і розмірів. Розгортання і згортання вікна. Керування прозорістю вікна. Модальні вікна.	Знати функції для роботи з вікнами (координати, розмір, прозорість), графічні віджети. Вміти створювати звичайні і модальні вікна, змінювати їх координати, розміри, прозорість, розміщувати у вікнах віджети і обробляти події).	Поточний контроль, тести, завдання для самостійної роботи

4. Система оцінювання курсу

Накопичування балів під час вивчення дисципліни	
Види навчальної роботи	Максимальна кількість балів
	3-й сем.
Лекції	15
Лабораторні роботи	30
Самостійна робота	5
Індивідуальне завдання	-
Залік/Екзамен	50
Максимальна кількість балів	100

5. Оцінювання відповідно до графіку навчального процесу

3-й семестр

Види навчальної	Номер навчального заняття (залежить від розподілу у розділі I)	Ра- зом
-----------------	--	------------

роботи	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	КСР	
Лекції	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		15
Лабораторні роботи	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		30
Самостійна робота																5	5
Індивідуальні завдання																	-
Екзамен																	50

6. Ресурсне забезпечення

Матеріально-технічне забезпечення	Інтерактивна панель, комп'ютерна лабораторія для лабораторних робіт з САПР для моделювання аналогових схем з кількістю комп'ютерів до 12, лабораторія електроніки з контрольно-вимірювальними приладами, блоками живлення, навчальними і макетними платами.
<p>Література:</p> <p>Основна</p> <p>1. Методичні вказівки "Використання Python, бібліотек NumPy та Matplotlib" до виконання самостійних робіт з навчальної дисципліни "Основи штучного інтелекту" [Електронний ресурс] : для студентів за спец. "Автоматизація, комп'ютерно-інтегровані технології та робототехніка" / уклад.: Д. В. Сальніков, В. А. Крилова ; Нац. техн. ун-т "Харків. політехн. ін-т". – Електрон. текст. дані. – Харків : НТУ "ХПІ", 2025. – 48 с. – URI: https://repository.kpi.kharkov.ua/handle/KhPI-Press/87441.</p> <p>2. Методичні вказівки до виконання лабораторних завдань з навчальної дисципліни "Програмування на Python" [Електронний ресурс] : для студентів денної та заочної форм навчання за спец. "122 Комп'ютерні науки", та "113 Прикладна математика" / уклад.: М. І. Шаповалова ; Нац. техн. ун-т "Харків. політехн. ін-т". – Електрон. текст. дані. – Харків, 2024. – 72 с. – URI: https://repository.kpi.kharkov.ua/handle/KhPI-Press/83306</p> <p>3. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник – К.: ВПЦ "Київський Університет", 2020. – 152 с.</p> <p>4. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник для студ. спеціальності 122 "Комп'ютерні науки", спеціалізації "Інформаційні технології в біології та медицині" / А. В. Яковенко ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,59 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 195 с.</p> <p>5. Олексій Васильєв. Програмування мовою Python. – Тернопіль: "Богдан", 2019 – 504 с.</p> <p>6. Програмування числових методів мовою Python: підруч. / А. В.</p>	

Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий; за ред. А. В. Анісімова. – К.: Видавничо-поліграфічний центр “Київський університет”, 2014. – 640 с.

7. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник для студ. спеціальності 122 "Комп'ютерні науки" / А. В. Яковенко ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,59 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 195 с.

Допоміжна

8. Eric Matthes. Python Crash Course, 3rd Edition: A Hands-On, Project-Based Introduction to Programming 3rd Edition. No Starch Press, 2023. – 552 p.

9. Vijay Kumar Sharma, Vimai Kumar, Swati Sharma, Shashwat Patnak. Python programming. A Practical Approach. CRC Press, 2022. - 309 p.

10. Luciano Ramalho. “Fluent Python. Clear, Concise, and Effective Programming. 2nd Edition”. O'reilly Media, 2022. – 1012 p.

11. Michal Jaworski, Tarek Ziade. Expert Python programming. Pact, 2021. – 611 p.

12. John Hant. Advanced guide to Python 3 programming. Springer, 2019, – 497.

Електронні ресурси

13. Навчальний ресурс <https://www.w3schools.com/>

14. Навчальний ресурс <https://realpython.com/>

15. Online інтерпретатор <https://www.jdoodle.com/python3-programming-online>

7. Контактна інформація

Кафедра	Комп'ютерної інженерії та електроніки, вул. Шевченка, 57, 210а, 59-60-07, https://kkite.pnu.edu.ua/ , kkie@pnu.edu.ua
Викладач (і) Гостьові лектори	Голота Віктор Іванович, к.т.н., доцент
Контактна інформація викладача	viktor.holota@cnu.edu.ua

8. Політика навчальної дисципліни

Академічна
добросесність

Дотримання академічної добросесності засновується на ряді положень та принципів академічної добросесності, що регламентують діяльність здобувачів вищої освіти та викладачів університету: <https://pnu.edu.ua/положення-про-запобігання-плагіату/>

Не допускається підказування та списування під час здачі будь-яких робіт поточного, рубіжного, підсумкового

	<p>контролю.</p> <p>Не допускається користування телефонами та будь-якими іншими електронними засобами під час здачі будь-яких робіт поточного, рубіжного, чи підсумкового контролю.</p>
<p>Пропуски занять (відпрацювання)</p>	<p>Можливість і порядок відпрацювання пропущених здобувачем освіти занять регламентується Положення про порядок організації та проведення оцінювання успішності здобувачів освіти ДВНЗ «Прикарпатського національного університету ім. Василя Стефаника» (введено в дію наказом ректора № 799 від 26.11.2019 р.; із внесеними змінами наказом № 212 від 06.04.2021 р.).</p>
<p>Виконання завдання пізніше встановленого терміну</p>	<p>У разі виконання завдання здобувачем освіти пізніше встановленого терміну, без попереднього узгодження ситуації з викладачем, оцінка виставляється», відповідно до Положення про порядок організації та проведення оцінювання успішності студентів ДВНЗ «Прикарпатського національного університету ім. Василя Стефаника» (введено в дію наказом ректора № 799 від 26.11.2019 р.; із внесеними змінами наказом № 212 від 06.04.2021 р.).</p>
<p>Невідповідна поведінка під час заняття</p>	<p>Невідповідна поведінка під час заняття регламентується рядом положень про академічну доброчесність та може призвести до відрахування здобувача вищої освіти «за порушення навчальної дисципліни і правил внутрішнього розпорядку вищого закладу освіти», відповідно до Положення про порядок переведення, відрахування та поновлення студентів вищих закладів освіти» (затверджене наказом Міністерства України № 245 від 15.07.1996 р.).</p>
<p>Додаткові бали</p>	<p>Студент має змогу також отримати додаткові бали, пройшовши навчальний курс у вигляді неформальної освіти з отриманням сертифікату в межах тематики дисципліни впродовж навчального семестру; взявши участь у науковому, освітньому чи прикладному проєкті, конференції, круглому столі, інших видах наукової активності, які відповідають профілю дисципліни; опублікувавши наукову працю, яка відповідає профілю дисципліни. Відповідно до Положення про порядок організації та проведення оцінювання успішності студентів ДВНЗ «Прикарпатського національного університету ім. Василя Стефаника» (введено в дію наказом ректора № 799 від 26.11.2019 р.; із внесеними змінами наказом № 212 від 06.04.2021 р.). відповідні студенти можуть отримати додаткові бали на підставі рішенням кафедри.</p>
<p>Неформальна освіта</p>	<p>У випадку, коли студент приймав участь у програмі мобільності, можливе врахування отриманих оцінок в іншому навчальному закладі за умови відповідності навчальних планів.</p> <p>Можливість зарахування результатів неформальної освіти регламентується Положенням про порядок зарахування</p>

результатів неформальної освіти у ДВНЗ «Прикарпатський національний університет імені Василя Стефаника» (введено в дію наказом ректора № 819 від 29.11.2019; із внесеними змінами наказом № 80 від 12.02.2021 р.).

Рекомендовані платформи:

<https://ua.udemy.com/>;

<https://www.coursera.org/>

<https://prometheus.org.ua/>



Голота В.І.