



Факультет математики та інформатики
Кафедра диференціальних рівнянь і прикладної математики

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ПРОГРАМУВАННЯ НА PYTHON

Рівень вищої освіти: Перший (бакалаврський)

Освітні програми: Комп'ютерне моделювання та
технології програмування

Спеціальність: F1 Прикладна математика

Галузь знань: F Інформаційні технології

Затверджено на засіданні кафедри
Протокол № 1 від 26 серпня 2025 р.

ЗМІСТ

1. [Загальна інформація](#)
2. [Анотація навчальної дисципліни](#)
3. [Мета та цілі навчальної дисципліни](#)
4. [Програмні компетентності](#)
5. [Програмні результати навчання](#)
6. [Структура навчальної дисципліни](#)
7. [Система оцінювання навчальної дисципліни](#)
8. [Політика навчальної дисципліни](#)
9. [Ресурсне забезпечення навчальної дисципліни](#)

1. ЗАГАЛЬНА ІНФОРМАЦІЯ

Назва дисципліни	Програмування на Python Programming with Python
Освітні програми	Комп'ютерне моделювання та технології програмування
Спеціальність	F1 Прикладна математика
Галузь знань	F Інформаційні технології
Освітній рівень	перший (бакалаврський)
Статус дисципліни	нормативна (з циклу професійної підготовки)
Рік навчання/Семестр	1 ^{ий} / 1 ^{ий}
Обсяг дисципліни	6 кредитів ECTS / 180 год.
Розподіл годин за видами занять	лекції – 26 год лабораторні – 48 год самостійна робота – 106 год
Форма контролю	екзамен
Мова викладання	українська
Посилання на сайт дистанційного навчання	https://d-learn.pnu.edu.ua/PwP https://classroom.google.com/PwP
Пререквізити	Інформатика
Постреквізити	ОК15 Алгоритми і структури даних ОК17 Об'єктно-орієнтоване програмування ОК24 Курсова робота (з програмування) ОК38 Практикум з моделювання і програмування

Кафедра	диференціальних рівнянь і прикладної математики ауд. 315 ЦК, https://kdrpm.cnu.edu.ua
Викладач(-і)	Віктор МАЗУРЕНКО
Контактний телефон	(0342)596027
E-mail	viktor.mazurenko@cnu.edu.ua
Профайл	https://mazurenko.cnu.edu.ua
Консультації	згідно з розкладом консультацій на сайті кафедри

2. АНОТАЦІЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

У курсі «Програмування на Python» здобувачі вищої освіти за спеціальністю «Прикладна математика» мають нагоду познайомитись з простою у засвоєнні і достатньо потужною для розв'язання прикладних задач мовою програмування Python, яка надає зручні високорівневі структури даних, а також простий, але ефективний підхід до об'єктно-орієнтованого програмування. Його лаконічний синтаксис і динамічна типізація разом з тим, що він є інтерпретованим і підтримує модульність, роблять Python чудовою мовою для написання сценаріїв та швидкої розробки додатків у різних областях (автоматизація задач, аналіз даних, графічний інтерфейс користувача, веб-розробка, розробка мобільних додатків, машинне навчання) і на більшості платформ (GNU/Linux, Windows, MacOS, FreeBSD, Solaris, OS/2, iOS, Android, Windows CE, PlayStation та інші).

3. МЕТА І ЦІЛІ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Набуття студентами загальних і фахових компетентностей з програмування для автоматизації задач у різних сферах людської діяльності.

Ознайомити студентів з основними поняттями, парадигмами і базовими структурами програмування, синтаксисом мови Python; сформувати у студентів уміння і навички написання програм мовою Python для автоматизації задач різної складності на основі розробки алгоритмів їх розв'язання.

4. ПРОГРАМНІ КОМПЕТЕНТНОСТІ

Відповідно до освітньо-професійної програми «Комп'ютерне моделювання та технології програмування» для першого (бакалаврського) рівня вищої освіти:

ІК. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми прикладної математики у професійній діяльності або у процесі навчання, що передбачає застосування математичних теорій та методів і характеризується комплексністю та невизначеністю умов.

ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.

ФК05. Здатність проектувати бази даних, інформаційні системи та ресурси.

ФК08. Здатність використовувати сучасні технології програмування та тестування програмного забезпечення.

5. ПРОГРАМНІ РЕЗУЛЬТАТИ НАВЧАННЯ

РН09. Будувати ефективні щодо точності обчислень, стійкості, швидкодії та витрат системних ресурсів алгоритми для чисельного дослідження математичних моделей та розв'язання практичних задач.

РН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символічних алгоритмів.

6. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ТЕМА, ПЛАН	КІЛЬКІСТЬ ГОДИН		
	ЛЕКЦІЇ	ЛАБОРАТОРНІ	САМОСТІЙНА РОБОТА
ПЕРШЕ ЗНАЙОМСТВО З МОВОЮ PYTHON - основні парадигми програмування та їх реалізація сучасними мовами - поняття алгоритму і програми, основні етапи побудови програми - компіляція та інтерпретація програмного коду - про автора розробки і офіційне представлення мови Python - короткий опис, сфери застосування і філософія мови - версії інтерпретатора Python і запуск інтерпретатора в інтерактивному режимі; інтегровані середовища розробки (IDLE)	4	2	4
ЛІНІЙНА СТРУКТУРА (ПОСЛІДОВНЕ ВИКОНАННЯ) ПРОГРАМИ - синтаксис мови Python: основні правила, ідентифікатори і ключові слова, константи і змінні, коментування програмного коду - поняття об'єкту, його ідентичність, тип і значення; змінювані (mutable) і незмінювані (immutable) об'єкти; - прості типи об'єктів (int, float, complex, str, bool), інтроспекція і перетворення типів - базові інструкції: пропустити (pass), присвоєння (=), введення (input), виведення (print) - послідовне виконання інструкцій, арифметичні операції над числами (int, float, complex) та їх пріоритет - як влаштовані дійсні і комплексні числа, використання функцій модулів math і cmath	4	4	8
УМОВНІ КЕРУЮЧІ КОНСТРУКЦІЇ - логічні операції і операції відношення, їх пріоритет - логічний тип об'єктів (bool): що істина, а що хибність - галуження як керуюча конструкція: умовний оператор if...else; каскадний умовний оператор if...elif...else; тернарний умовний оператор ...if...else...	2	4	8
ЦИКЛІЧНІ КЕРУЮЧІ КОНСТРУКЦІЇ - цикл як керуюча конструкція: цикл з передумовою while, цикл по колекції for...in, переривання і продовження циклів - арифметична прогресія як колекція range, поняття про інкремент і декремент - рекурентні співвідношення та їх системи - числові послідовності та їх границі	4	6	10
СПИСКИ І КОРТЕЖІ - списки (list): створення, індексація, зрізи, змінюваність списків - операції над списками: конкатенація і зчеплення, перевірка приналежності до списку, аналіз списку, модифікація списку - введення списків з клавіатури, функція map() - обхід списку: за індексами, за елементами, через функції enumerate() і zip(); матриці як списки зі списків	2	6	8

<ul style="list-style-type: none"> - кортежі (tuple): створення, індексація, зрізи, незмінюваність кортежів - операції над кортежами: аналогічні до операцій над списками, пакування і розпакування кортежів, обмін значеннями змінних - генератори послідовностей (зокрема списків і кортежів), методи <code>iter()</code> і <code>next()</code>, функція <code>sum()</code> 			
<p>СИМВОЛИ І РЯДКИ</p> <ul style="list-style-type: none"> - символи і таблиці кодування, спеціальні (екрановані) символи, операції із символами - рядок як кортеж символів, індексація, зрізи, довжина рядка, незмінюваність рядків - операції над рядками: конкатенація і зчеплення, перевірка приналежності до рядка, пошук і заміна, аналіз рядка, модифікація рядка, розбиття і побудова рядка - порівняння і форматування рядків, метод <code>format()</code> і f-рядки 	2	4	8
<p>СЛОВНИКИ І МНОЖИНИ</p> <ul style="list-style-type: none"> - словники (dict): створення і доступ до елементів, невпорядкованість і незмінюваність словників - операції над словниками: перевірка приналежності до словника, аналіз словника, модифікація списку - обхід словника: за ключами, за значеннями, за парами «ключ-значення» - множини (set): створення, індексація, зрізи, невпорядкованість і змінюваність множин - операції над множинами: операції відношення; перетин, об'єднання, різниця і симетрична різниця; додавання і видалення елементів - незмінювані множини (frozenset) 	2	6	10
<p>ФУНКЦІЇ</p> <ul style="list-style-type: none"> - поняття підпрограми і функції, визначення і виклик функцій, повернення кількох значень - аргументи (параметри) функції: формальні і фактичні, позиційні та іменовані (ключові) аргументи; типові (за замовчуванням) значення аргументів - функції з довільною кількістю аргументів, області видимості: локальні і глобальні змінні - поняття про функційний тип даних, анонімні (lambda-функції) - рекурсія і рекурсивні функції, генератор-функції - декоратори для функцій: означення, загальний опис, вкладені декоратори 	2	6	10
<p>ОБРОБКА ПОМИЛОК І ВИНЯТКІВ</p> <ul style="list-style-type: none"> - помилки у програмному кодї та їх типи: синтаксичні, логічні і помилки виконання - виняткові ситуації, обробка винятків з допомогою <code>try...except</code>, стандартні класи винятків, створення та ініціювання користувацьких винятків - менеджер контексту <code>with...as</code> 	2	4	4
<p>ФАЙЛИ І МОДУЛІ</p> <ul style="list-style-type: none"> - файл і його ознаки, типи файлів - етапи роботи з файлом: відкриття, читання і запис, закриття; текстові файли - бінарні файли і серелізатори - робота з файловою системою 	2	4	6

- імпорт і використання модулів, головний модуль програми - розташування модулів у файловій системі - пакет і його структура			
МОДУЛЬНИЙ КОНТРОЛЬ	–	2	12
ПІДСУМКОВИЙ КОНТРОЛЬ (ЕКЗАМЕН)	–	–	18
ВСЬОГО	26	48	106

7. СИСТЕМА ОЦІНЮВАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Загальна система оцінювання	Підсумкова оцінка з навчальної дисципліни є сумою оцінок з таких видів навчальних робіт: лабораторні роботи (виконання і захист), модульний контроль (тест), самостійна робота (за окремими темами) та підсумковий контроль (екзамен). Підсумкова оцінка визначається відповідно до поданої нижче таблиці оцінювання за різними шкалами (100-бальна, ECTS, національна).
Лабораторні роботи	Максимальна оцінка за вчасно виконані і захищені лабораторні роботи (вісім з яких є індивідуальними, а одна – командною) становить 30 балів.
Модульний контроль	Тест з програмування на Python містить 15 завдань закритого/відкритого типу на знання парадигм програмування, на розуміння синтаксису і базових конструкцій мови, на вміння працювати з простими типами даних і структурами даних, на застосування функцій зі стандартних модулів і створення функцій розробника, на тестування програмного коду і обробку помилок. Максимальна оцінка за тест становить 15 балів.
Самостійна робота	Максимальна оцінка за активну і змістовну аудиторну роботу та самостійну роботу за окремими темами становить 5 балів.
Умови допуску до підсумкового контролю	Загальна кількість балів за всі види робіт впродовж семестру становить не менше 25.
Підсумковий контроль	Екзамен складається з 2-х частин: тест на знання теоретичних основ програмування мовою Python та написання програми на Python для вирішення 2-х практичних задач. Максимальна оцінка за екзамен становить 50 балів.

Накопичування балів впродовж вивчення дисципліни

Види навчальної роботи	Максимальна кількість балів
Лабораторні роботи	30
Модульний контроль	15
Самостійна робота	5
Підсумковий контроль	50
Підсумкова оцінка	100

ШКАЛА ОЦІНЮВАННЯ: НАЦІОНАЛЬНА ТА ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проєкту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
80 – 89	B	добре	
70 – 79	C		
60 – 69	D	задовільно	
50 – 59	E		
26 – 49	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-25	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

КРИТЕРІЇ ОЦІНЮВАННЯ НАВЧАЛЬНИХ ДОСЯГНЕНЬ

(відповідно до [Положення](#) про організацію освітнього процесу)

«відмінно» – здобувач освіти міцно засвоїв теоретичний матеріал, глибоко і всебічно знає зміст навчальної дисципліни, основні положення рекомендованої літератури, логічно мислить і буде відповідь, вільно використовує набуті теоретичні знання при аналізі практичного матеріалу, висловлює своє ставлення до тих чи інших проблем, демонструє високий рівень засвоєння практичних навичок;

«добре» – здобувач освіти добре засвоїв теоретичний матеріал, володіє основними аспектами рекомендованої літератури, аргументовано викладає його; має практичні навички, висловлює свої міркування з приводу тих чи інших проблем, але припускається певних неточностей і похибок у логіці викладу теоретичного матеріалу або при аналізі практичного матеріалу;

«задовільно» – здобувач освіти в основному опанував теоретичними знаннями навчальної дисципліни, орієнтується в першоджерелах та рекомендованій літературі, але непереконливо відповідає, плутає поняття, додаткові питання викликають невпевненість або відсутність стабільних знань; відповідаючи на запитання практичного характеру, виявляє неточності у знаннях, не вміє оцінювати факти та явища, пов'язувати їх із майбутньою діяльністю;

«незадовільно» – здобувач освіти не опанував навчальний матеріал дисципліни, не знає наукових фактів, визначень, майже не орієнтується в рекомендованій літературі, відсутнє наукове мислення, практичні навички не сформовані.

8. ПОЛІТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Академічна добросесність	Важливим є самостійне виконання навчальних завдань та посилання на джерела інформації у разі використання ідей, розробок, тверджень, відомостей. Плагіат та інші види академічної недобросесності не принесуть користі, тому є недоречними та контролюються відповідно до Положення
Пропуски занять (відпрацювання)	Наслідком періодичних пропусків є самостійне опрацювання навчального матеріалу з можливим консультуванням у викладача відповідно до графіка консультацій. Наслідком систематичних пропусків є додаткові види самостійної роботи, які контролюються на передбачених у графіку навчального процесу тижнях контролю самостійної роботи

Виконання завдань пізніше встановленого терміну	Приводить до втрати частини балів, запланованих у системі оцінюванні навчальної дисципліни (усі види навчальної роботи важливо виконувати належним чином і вчасно, щоб зберігати загальний темп курсу, котрий сприяє ефективному засвоєнню матеріалу без шкоди здоров'ю
Додаткові бали	До 5 балів за активність, комунікативність, креативність, наполегливість, самостійність при вивченні дисципліни
Неформальна освіта	Можливість повного або часткового зарахування результатів неформальної освіти відповідно до Положення . Рекомендовані платформи: coursera.org , prometheus.org.ua , udemy.com , netacad.com , edx.org

9. РЕСУРСНЕ ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Матеріально-технічне забезпечення	Лекційна аудиторія, мультимедіа, комп'ютерна лабораторія з програмним забезпеченням
Навчально-методичне забезпечення	Навчальний контент на освітніх платформах https://d-learn.pnu.edu.ua/PwP https://classroom.google.com/PwP
Інформаційне забезпечення	Наукова бібліотека КНУВС , Рекомендована література

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Васильєв О.М. Програмування мовою Python. - Тернопіль: Навчальна книга - Богдан, 2019. – Режим доступу: [url](#)
2. Висоцька В.А., Оборська О.В. Python: алгоритмізація та програмування: навчальний посібник. – Львів: Новий Світ - 2000, 2021. – Режим доступу: [url](#)
3. Крєневич А.П. Python у прикладах і задачах. Частина 1. Структурне програмування: Навчальний посібник. - К.: ВПЦ "Київський університет", 2017. – Режим доступу: [url](#)
4. Мазуренко В.В., Дмитришин М.І., Василичин П.Б. Структурне програмування з Python: Лабораторний практикум. – Ів.-Фр.: Голіней, 2023.
5. Мізюк О. Путівник мовою програмування Python. – Ел. вид., 2020. – Режим доступу: [url](#)
6. Селіверстов Р., Мельничин А. Основи програмування мовою Python: навч. посібник. – Львів: ЛНУ імені Івана Франка, 2020.
7. Severance Ch.R. Python for Everybody: Exploring Data Using Python 3, 2023. – Access mode: [url](#)
8. Swaroop С.Н. A byte of Python, 2005. – Access mode: [url](#)
9. PEP 8 - Style Guide for Python Code. – Access mode: [url](#)
10. The Python Tutorial. – Access mode: [url](#)

Викладач В.Мазуренко