

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**  
**Кафедра програмування**

**Затверджено**

На засіданні кафедри програмування  
факультету прикладної математики та  
інформатики Львівського національного  
університету імені Івана Франка  
(протокол № 1 від 29 серпня 2023 р.)



---

Зав. кафедри: к. ф.-м. н., доц. Ярошко С. А.

**СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**  
**“Програмування” (ІІІ семестр),**  
**що викладається в межах ОПП “Інформатика”**  
**першого (бакалаврського) рівня вищої освіти**  
**для здобувачів зі спеціальності 122 “Комп’ютерні науки”**

<b>Назва дисципліни</b>	Програмування.
<b>Адреса викладання дисципліни</b>	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики, кафедра програмування
<b>Галузь знань, шифр та назва спеціальності</b>	Галузь знань: 12 Інформаційні технології Спеціальність: 122 Комп'ютерні науки
<b>Викладачі дисципліни</b>	Музичук Анатолій Омелянович, к. ф.-м. н., доц., доцент кафедри програмування Селіверстов Роман Григорович, к. ф.-м. н., доц., доцент кафедри програмування
<b>Контактна інформація викладачів</b>	<i>Електронна пошта:</i> <a href="mailto:anatoliy.muzychuk@lnu.edu.ua">anatoliy.muzychuk@lnu.edu.ua</a> <a href="mailto:roman.seliverstov@lnu.edu.ua">roman.seliverstov@lnu.edu.ua</a> <i>Вебсторінки:</i> <a href="https://ami.lnu.edu.ua/employee/muzychuk">https://ami.lnu.edu.ua/employee/muzychuk</a> <a href="https://ami.lnu.edu.ua/employee/seliverstov-roman-hryhorovych">https://ami.lnu.edu.ua/employee/seliverstov-roman-hryhorovych</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Один раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі онлайн консультації в середовищі Microsoft Teams. Для погодження часу онлайн консультацій писати на електронну пошту викладача.
<b>Сторінка курсу</b>	<a href="https://ami.lnu.edu.ua/course/prohramuvannia-chastyna-2-python">https://ami.lnu.edu.ua/course/prohramuvannia-chastyna-2-python</a>
<b>Інформація про дисципліну</b>	Курс "Програмування" є нормативною навчальною дисципліною зі спеціальності 122 "Комп'ютерні науки (Інформатика)" для освітньої програми "Інформатика". У третьому семестрі її викладають в обсязі 4 кредитів (за Європейською кредитно-трансферною системою ECTS)
<b>Коротка анотація дисципліни</b>	Дисципліна "Програмування" фокусується на особливостях моделі даних та інших тонкощах мови Python 3. Значна увага звертається на аналіз та візуалізацію даних засобами спеціалізованих бібліотек.
<b>Мета та цілі дисципліни</b>	Мета – формування компетенцій, необхідних для ефективного використання бібліотек мови Python для вирішення прикладних професійних та наукових задач.  Цілі: ознайомлення з технологіями опрацювання та візуалізації даних засобами бібліотек мови Python, оволодіння навичками складання та програмування алгоритмів розв'язання прикладних задач у імперативному, процедурному та об'єктно-орієнтованому

	СТИЛЯХ.
<b>Література для вивчення дисципліни</b>	<p><i>Основна</i></p> <ol style="list-style-type: none"> <li>1. <a href="#">Матгес Е. Пришвидшений курс Python.</a> – Львів : ВСЛ, 2021 – 600 с.</li> <li>2. Селіверстов Р., Мельничин А. Основи програмування мовою Python: навч. посібник. – Львів : ЛНУ імені Івана Франка, 2020 – 190 с.</li> <li>3. Васильєв О. Програмування в PYTHON. Теорія і практика. – К. : Ліра-К, 2023. – 462 с.</li> <li>4. Lambert K. A. Fundamentals of Python: First Programs, 2nd Edition. – Cengage, 2019.</li> <li>5. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 2nd Edition. – O'Reilly Media, 2018.</li> <li>6. The Python Tutorial. – <a href="https://docs.python.org/3/tutorial/index.html">https://docs.python.org/3/tutorial/index.html</a></li> <li>7. NumPy. – <a href="http://numpy.org">http://numpy.org</a>.</li> <li>8. Pandas. – <a href="http://pandas.pydata.org">http://pandas.pydata.org</a>.</li> <li>9. Matplotlib. – <a href="http://matplotlib.org">http://matplotlib.org</a>.</li> </ol> <p><i>Додаткова</i></p> <ol style="list-style-type: none"> <li>10. David M. Beazley. Python Distilled. – Addison-Wesley Professional, 2021.</li> <li>11. Wentworth P., Elkner J., Downey A., Meyers C. How to Think Like a Computer Scientist: Learning with Python 3. – Green Tea Press, 2018.</li> <li>12. Python Tricks: The Book. — Dan Bader, 2017.</li> <li>13. Sweigart A. Automate the Boring Stuff with Python: Practical Programming for Total Beginners. – No Starch Press, 2014.</li> </ol>
<b>Обсяг курсу</b>	4 кредити ЄКТС (120 годин). З них 32 години лекцій, 32 години лабораторних занять, 56 годин самостійної роботи.
<b>Очікувані результати навчання</b>	<p>Після завершення цього курсу студент буде <i>знати</i>:</p> <ul style="list-style-type: none"> <li>– синтаксис, основні конструкції та типи даних мови Python;</li> <li>– принципи імперативного, процедурного, функціонального та об'єктно-орієнтованого програмування мовою Python;</li> <li>– класичні та спеціалізовані стосовно виконання у середовищі Python алгоритми обробки та візуалізації даних, які реалізовані у популярних бібліотеках.</li> </ul> <p><i>вміти</i>:</p> <ul style="list-style-type: none"> <li>– реалізовувати класичні алгоритми мовою Python;</li> <li>– використовувати переваги об'єктно-орієнтованого програмування під час написання програм;</li> <li>– створювати багатомодульні програми;</li> <li>– розробляти графічні інтерфейси;</li> <li>– представляти, структурувати, обробляти та візуалізувати дані у комп'ютерних системах;</li> <li>– застосовувати отримані знання для розв'язання конкретних професійних задач.</li> </ul>

<p><b>Компетентності</b></p>	<p><i>Інтегральна (ІК):</i> Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачають застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.</p> <p><i>Загальні (ЗК):</i>  ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.  ЗК2. Здатність застосовувати знання у практичних ситуаціях.  ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.  ЗК8. Здатність генерувати нові ідеї (креативність).  ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.</p> <p><i>Спеціальні (фахові, предметні) компетентності (СК):</i>  СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.  СК7. Здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів.  СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.  СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.</p>
<p><b>Програмні результати навчання</b></p>	<p>ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.</p> <p>ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.</p> <p>ПР6. Використовувати методи чисельного диференціювання та інтегрування функцій, розв'язання звичайних диференціальних та інтегральних рівнянь, особливостей чисельних методів та можливостей їх адаптації до інженерних задач, мати навички програмної реалізації чисельних методів.</p> <p>ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості</p>

	застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук. ПР15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.				
<b>Ключові слова</b>	алгоритм, скрипт, тип даних, IDLE, інструкція, функція, об'єкт, клас, поліморфізм, наслідування, інкапсуляція				
<b>Формат курсу</b>	Очний				
<b>Теми</b>	<b>Тижень</b>	<b>Тема, план, короткі тези</b>	<b>Форма діяльності</b>	<b>Завдання, год</b>	<b>Термін виконання</b>
	1	<b>Модель даних, базові типи, оператори та інструкції Python.</b> <i>Режими виконання коду. Основні прийоми роботи в IDLE та Jupyter Notebook. Коментування коду, довідка Python. Особливості числових типів. Введення та виведення даних. Стандартна бібліотека Python. Імпорт. [1-4, 6, 10, 13]</i>	лекція	2	
		<i>Програмна реалізація лінійних алгоритмів</i>	лабораторне заняття	2	у межах заняття
		<i>Особливості і наслідки динамічної типізації. Рівність та ідентичність об'єктів. Кежування рядків і чисел. [6, 10, 12]</i>	самотійна робота	4	
	2	<b>Розгалуження та цикли.</b> <i>Розгалуження в алгоритмах. Логічні вирази. Конструкції if та match. Функція range(). Цикли for і while. Операції break і continue. Блок else. [1-4, 6, 10, 13]</i>	лекція	2	
		<i>Програмна реалізація алгоритмів з розгалуженнями та циклами</i>	лабораторне заняття	2	наступне лаб. заняття
		<i>Зіставлення із взірцем (Pattern matching). Основи ітерування по</i>	самотійна робота	3	

		колекціях. [6, 10, 12]			
3	<b>Рядки та файли:</b> Літерали рядків. Символи. Керівні символи. Індекси та зрізи. Конкатенація та повторення рядків. Форматування рядків. Функції для роботи з рядками та методи рядків. Перевірка на входження та посимвольний обхід рядка. Режими відкриття файлів. Операції з файлами. Інструкція with/as. Абсолютні та відносні шляхи. [1-4, 6, 10, 13]	лекція	2		
	Програмування з використанням рядків і файлів	лабораторне заняття	2	наступне лаб. заняття	
	Байтові потоки. Модулі sys і os [4, 6, 11, 12, 13]	самостійна робота	4		
4	<b>Колекції:</b> Списки. Кортежі. Словники. Множини. Доступ до елементів. Перевірка на входження та поелементний обхід. Основні функції та методи. Генерування колекцій. [1-4, 6, 10, 13]	лекція	2		
	Програмування з використанням колекцій	лабораторне заняття	2	наступне лаб. заняття	
	Модуль collections для контейнерних типів даних [6, 10]	самостійна робота	3		
5	<b>Процедурне програмування.</b> Декомпозиція. Функції. Створення та виклик функцій. Інструкція return. Області видимості. Локальні, нелокальні та глобальні змінні. Позиційні та іменовані аргументи. Значення за замовчуванням. Передавання довільної кількості аргументів.	лекція	2		

	<p><i>Розпаковування аргументів. Рекурсивні функції. Лямбда-функції. Функції zip, map, filter, all, any. [1-4, 6, 10, 11, 12, 13]</i></p>			
	<p><i>Процедурне програмування. Декомпозиція</i></p>	лабораторне заняття	2	наступне лаб. заняття
	<p><i>Спеціальні режими зіставлення аргументів. Анотування типів [1, 2, 4, 6, 10, 12]</i></p>	самотійна робота	3	
6-9	<p><b>ООП.</b> Класи та екземпляри. Атрибути та методи, особливості інкапсуляції. Методи <code>__init__()</code> та <code>__str__()</code>. Особливості керування доступом до атрибутів; декоратор <code>property</code>. Змінні та методи класу. Статичні методи. Перевантаження операторів. Наслідування. Особливості множинного наслідування. Перевизначення методів. Виклик методів базового класу. Композиція. [1-4, 6, 10]</p>	лекція	4	
	<p><i>Модифікатори доступу. Геттери та сеттери. Вбудовані атрибути. Слот. Абстрактні класи та методи. [1-4, 6, 10]</i></p>	самотійна робота	9	
	<p><b>Обробка винятків.</b> Ієрархія класів винятків. Інструкції <code>try</code>, <code>assert</code>, <code>raise</code>. [1-4, 6, 10]</p>	лекція	2	
	<p><b>Тестування програм.</b> Модулі <code>doctest</code> і <code>unittest</code> [1, 4, 6, 10, 11]</p>	лекція	2	
	<p><i>Об'єктно-орієнтоване програмування та модульне тестування</i></p>	лабораторне заняття	8	5 тижнів
	<p><i>Документування функцій та класів. Тестування винятків [1, 3, 6, 11]</i></p>	самотійна робота	5,5	

	10-11	<b>Прийоми ефективного програмування.</b> Функція як об'єкт. Атрибути функції. Функції-декоратори. Класи-декоратори. Декорування функцій та класів. Прийоми ефективного використання вбудованих колекцій. Ітератори. Генератори. Менеджери контексту. [1, 2, 6, 10]	лекція	4	
		Прийоми ефективного програмування	лабораторне заняття	4	3 тижні
		Удосконалені генератори (Enhanced generators). Декоратор @contextmanager [6, 10, 12]	самотійна робота	7	
	12	<b>Основи програмування графічних інтерфейсів.</b> Графічний інтерфейс користувача. Модуль tkinter. Базове вікно. Елементи управління (віджети). Обробка подій. [1-4, 6, 13]	лекція	2	
		Програмування графічних інтерфейсів	лабораторне заняття	2	наступне лаб. заняття
		Методи розташування віджетів (grid, pack, place) [1-4, 6, 13]	самотійна робота	3,5	
	13	<b>Інтерактивний комп'ютинг.</b> Візуалізація даних. Робота в середовищі Jupyter Notebook. Віджети. Візуалізація даних засобами matplotlib. [2, 5, 9, 11]	лекція	2	
		Створення інтерактивних ноутбуків	лабораторне заняття	2	наступне лаб. заняття
		Типи діаграм та графіків matplotlib [2, 5, 9]	самотійна робота	3,5	
	14	<b>Бібліотека NumPy.</b> [5, 7]	лекція	2	



		<i>Матрично-векторні обчислення засобами NumPy.</i>	<i>лабораторне заняття</i>	2	<i>наступне лаб. заняття</i>
		<i>Лінійна алгебра в NumPy [5, 7]</i>	<i>самостійна робота</i>	3,5	
15		<i>Аналіз гетерогенних даних засобами Pandas. Типи Series і DataFrame. Конкатенація та злиття даних. Фільтрування, сортування, групування. [5, 8, 9, 13]</i>	<i>лекція</i>	2	
		<i>Первинний статистичний аналіз наборів даних.</i>	<i>лабораторне заняття</i>	2	<i>наступне лаб. заняття</i>
		<i>Інтерактивні запити та візуалізація статистичних даних з використанням віджету interact [5, 8, 9]</i>	<i>самостійна робота</i>	3	
16		<i>Імпорт C++ коду. Підсумки [6, 12]</i>	<i>лекція</i>	2	
		<i>Підсумкове заняття</i>	<i>лабораторне заняття</i>	2	<i>у межах заняття</i>
		<i>Модулі ctypes і array [6, 12]</i>	<i>самостійна робота</i>	4	
<b>Підсумковий контроль, форма</b>	Іспит у кінці семестру				
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з дисциплін "Програмування. Частина 1" і "Алгоритми і структури даних". Одночасно з вивченням курсу студенти проходять навчальну обчислювальну практику, впродовж якої виконують завдання з програмування, поглиблюють набуті в курсі знання та удосконалюють навички.				
<b>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</b>	Лекції з мультимедійними презентаціями; лабораторні заняття у вигляді виконання практичних завдань (у тому числі командних); самостійне опрацювання навчальних матеріалів, розміщених у хмарних сховищах (Moodle, Microsoft Teams); обговорення тем та консультації в середовищах Microsoft Teams, Slack, Skype тощо.				
<b>Необхідне обладнання</b>	Для проведення лекцій: комп'ютер, проєктор, доступ до мережі Інтернет. Для проведення лабораторних занять та виконання завдань: комп'ютер, ОС Windows/Linux, доступ до мережі Інтернет; прикладне програмне забезпечення: інтерпретатор Python 3.x, Python IDLE.				

<p><b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b></p>	<p>Оцінювання проводиться за 100-бальною шкалою. 50 балів нараховують за виконання лабораторних робіт, ще 50 балів – за виконання екзаменаційного завдання. Лабораторні завдання можуть бути індивідуальні та командні. Упродовж семестру студент виконує 10 лабораторних робіт / завдань, кожен(е) з яких оцінюють від 2 до 10 балів залежно від складності. Оцінка за екзаменаційне завдання поділена на дві частини: 20 балів за засвоєння теоретичного матеріалу (тест) та 30 балів за написання комп'ютерної програми (декількох програм). У екзаменаційному завданні прописані кількість балів за правильну відповідь на кожне питання теоретичної частини і правильну програмну реалізацію кожного структурного елемента завдання практичної частини.</p> <p>Студенти зобов'язані дотримуватися усіх строків, визначених для виконання усіх видів завдань, передбачених курсом. Звіти про виконання лабораторних робіт завантажують у відповідне хмарне сховище. Викладач має право затребувати захист лабораторної роботи.</p> <p>Очікується, що роботи студентів будуть їхнім оригінальними дослідженнями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих звітів становлять, але не обмежують приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p> <p>Жодні форми порушення академічної доброчесності не толеруються</p>
<p><b>Типові питання</b></p>	<p>Тематика теоретичної (тестової) частини:</p> <ol style="list-style-type: none"> <li>1. Базові типи Python.</li> <li>2. Особливості та наслідки динамічної типізації.</li> <li>3. Операції над числовими типами.</li> <li>4. Операції над рядками.</li> <li>5. Взаємодія з файлами.</li> <li>6. Колекції об'єктів.</li> <li>7. Умовні інструкції.</li> <li>8. Цикли.</li> <li>9. Оголошення і виклик функцій.</li> <li>10. Області видимості.</li> <li>11. Режими співставлення аргументів.</li> <li>12. Анонімні функції.</li> <li>13. Оперування функціями як об'єктами.</li> <li>14. Засоби функціонального програмування в Python.</li> <li>15. Модулі та пакети, операції імпорту.</li> <li>16. Класи і екземпляри.</li> <li>17. Модифікатори доступу.</li> <li>18. Перевантаження операторів.</li> <li>19. Наслідування та композиція.</li> <li>20. Ітератори та генератори.</li> <li>21. Обробка винятків та тестування коду.</li> <li>22. Декораторування функцій і класів.</li> </ol>

	<p>23. Векторно-матричне числення в NumPy.</p> <p>24. Первинний статистичний аналіз і візуалізація даних.</p> <p>25. Віджети та обробка подій у графічних інтерфейсах.</p> <p>Для успішного складання практичної частини (написання і виконання коду) потрібно вміти:</p> <ol style="list-style-type: none"> <li>1. Використовувати інструкції галуження та циклів, зокрема, з інструкціями <code>break</code>, <code>continue</code> і блоком <code>else</code> в циклах.</li> <li>2. Використовувати стандартні контейнери: кортежі, списки, словники, множини.</li> <li>3. Оголошувати та використовувати функції з позиційними та іменованими аргументами, у тому числі і зі змінною кількістю аргументів і значеннями за замовчуванням.</li> <li>4. Оперувати функціями як об'єктами.</li> <li>5. Використовувати лямбда-функції.</li> <li>6. Оголошувати та використовувати декоратори функцій.</li> <li>7. Оголошувати класи, визначаючи конструктори, методи, статичні атрибути, переважуючи оператори.</li> <li>8. Наслідувати класи, перевизначати методи.</li> <li>9. Створювати ітератори та генератори.</li> <li>10. Оголошувати, запускати та опрацьовувати винятки, розпізнаючи тип винятку та гарантуючи виконання певних дій незалежно від виникнення винятків.</li> <li>11. Взаємодіяти з файловою системою комп'ютера.</li> <li>12. Будувати основні типи графіків засобами бібліотеки <code>matplotlib</code>, налаштовувати їхні параметри.</li> <li>13. Використовувати засоби бібліотеки <code>numpy</code> для опрацювання числових масивів.</li> <li>14. Створювати Jupyter-ноутбуки з розміткою для форматування тексту та набору математичних формул, записувати та виконувати фрагменти програм, використовувати графічні елементи керування (<code>widgets</code>) для створення інтерактивного інтерфейсу користувача.</li> <li>15. Використовувати засоби бібліотеки <code>pandas</code> для опрацювання таблиць гетерогенних даних (завантаження таблиць, об'єднання таблиць, заповнення пропусків, вибірка, фільтрування, використання методів агрегації).</li> </ol>
<b>Опитування</b>	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.