

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 з навчальної дисципліни
“ІНФОРМАТИКА ТА ПРОГРАМУВАННЯ (PYTHON)”,
що викладається в межах ОПП “Статистичний аналіз даних”
першого (бакалаврського) рівня вищої освіти
для здобувачів зі спеціальності 112 “Статистика”

Назва дисципліни	Інформатика та програмування (Python)
Адреса викладання дисципліни	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики, кафедра програмування
Галузь знань, шифр та назва спеціальності	Галузь знань: 11 Математика та статистика Спеціальність: 112 Статистика
Викладачі дисципліни	Селіверстов Роман Григорович, к. ф.-м. н., доц., доцент кафедри програмування
Контактна інформація викладачів	<i>E-mail:</i> roman.seliverstov@lnu.edu.ua <i>Вебсторінка:</i> https://ami.lnu.edu.ua/employee/seliverstov-roman-hryhorovych
Консультації з питань навчання по дисципліні відбуваються	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.
Сторінка курсу	https://ami.lnu.edu.ua/course/informatyka-ta-prohramuvannia-python-112-statystyka
Інформація про дисципліну	Курс "Інформатика та програмування (Python)" є нормативною навчальною дисципліною зі спеціальності 112 "Статистика" для освітньої програми "Статистичний аналіз даних", яка викладається у першому і другому семестрах в обсязі 8 кредитів (за Європейською кредитно-трансферною системою ECTS).
Коротка анотація дисципліни	Дисципліна "Інформатика і програмування (Python)" забезпечує базову професійну підготовку з програмування мовою Python 3 бакалаврів статистики. Курс фокусується на особливостях моделі даних та інших тонкощах мови Python. Значна увага звертається на аналіз та візуалізацію даних засобами спеціалізованих бібліотек.
Мета та цілі дисципліни	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.

<p>Література для вивчення дисципліни</p>	<ol style="list-style-type: none"> 1. Маттес Е. Пришвидшений курс Python. – Львів : ВСЛ, 2021 – 600 с. 2. Селіверстов Р., Мельничин А. Основи програмування мовою Python: навч. посібник. – Львів : ЛНУ імені Івана Франка, 2020 – 190 с. 3. Lambert K. A. Fundamentals of Python: First Programs, 2nd Edition. – Cengage, 2019. 4. The Python Tutorial. – https://docs.python.org/3/tutorial/index.html 5. NumPy. – http://numpy.org. 6. Pandas. – http://pandas.pydata.org. 7. Matplotlib. – http://matplotlib.org.
<p>Обсяг курсу</p>	<p>Загальний обсяг: 240 годин. Аудиторних занять: 128 год., з них 64 години лекцій, 64 годин лабораторних занять, 112 годин самостійної роботи.</p>
<p>Очікувані результати навчання</p>	<p>У результаті вивчення даної навчальної дисципліни студент буде <i>знати</i>:</p> <ul style="list-style-type: none"> – синтаксис, основні конструкції та типи даних мови Python; – принципи імперативного, процедурного, функціонального та об'єктно-орієнтованого програмування мовою Python; – класичні та спеціалізовані стосовно виконання у середовищі Python алгоритми обробки та візуалізації даних, які реалізовані у популярних бібліотеках. <p><i>вміти</i>:</p> <ul style="list-style-type: none"> – реалізовувати класичні алгоритми мовою Python; – використовувати переваги об'єктно-орієнтованого програмування під час написання програм; – створювати багатомодульні програми; – розробляти графічні інтерфейси; – представляти, структурувати, обробляти та візуалізувати дані у комп'ютерних системах; – застосовувати одержані знання для розв'язання конкретних професійних задач. <p>В результаті засвоєння матеріалу даного курсу студент набуде таких загальних (ЗК) і спеціальних (фахових) (СК) компетентностей:</p> <p>ЗК-2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК-3. Знання й розуміння предметної області та професійної діяльності.</p> <p>ЗК-4. Здатність спілкуватися українською мовою як усно, так і письмово.</p> <p>ЗК-5. Здатність спілкуватися іноземною мовою.</p> <p>ЗК-6. Навички використання інформаційних і комунікаційних технологій.</p> <p>ЗК-7. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>ЗК-8. Здатність до пошуку, обробки та аналізу інформації з різних джерел.</p> <p>ЗК-10. Здатність працювати в команді.</p>

ЗК-11. Здатність до професійного спілкування з представниками інших професійних груп різного рівня (з експертами в інших галузях знань).

СК-2. Здатність застосовувати у професійній діяльності знання та навички в галузях теорії ймовірностей, математичної статистики, теорії випадкових процесів.

СК-4. Здатність до математичного формулювання задач та вибору методів їх розв'язання.

СК-6. Здатність до ймовірнісного мислення, що передбачає сприйняття стохастичної природи явищ.

СК-7. Здатність робити якісні висновки з кількісних даних.

СК-8. Уміння працювати з інформаційними базами даних.

СК-9. Здатність розробляти експериментальні та спостережувальні дослідження та аналізувати дані цих досліджень.

СК-10. Здатність проводити дослідження ймовірнісно-статистичних моделей та інтерпретувати одержані результати.

СК-11. Здатність використання обчислювальної техніки, спеціалізованих мов програмування та програмних засобів для розв'язання задач і здобуття додаткової інформації.

СК-12. Здатність застосовувати ймовірнісно-статистичні методи в міждисциплінарному контексті.

СК-13. Здатність подавати статистичні процедури та результати їхнього застосування у формі, придатній для цільової аудиторії, до якої звертаються, як усно, так і письмово.

СК-15. Здатність аналізувати основи і властивості базових економічних та фінансових структур, інтерпретувати показники фінансової діяльності, користуватися методами оптимального керування економічних та природних процесів.

СК-16. Здатність застосовувати у професійній діяльності знання та навички з машинного навчання, обробки зображень і природної мови.

СК-17. Здатність моделювати та пояснювати дані просторових і часових вибірок за допомогою знань і навичок з регресійного аналізу.

і здобуде такі **результати навчання (РН)**:

РН-4. Вміти пояснювати математичні концепції та статистичні методи мовою, зрозумілою для нефакхівців у галузі математики та статистики.

РН-12. Вміти збирати та обробляти дані, застосовувати статистичні процедури для аналізу даних за допомогою обчислювальної техніки та програмних засобів.

РН-14. Володіти сучасними інформаційними технологіями для створення презентацій, роботи з базами даних, пошуку інформації та обміну нею.

РН-16. Вміти використовувати в практичній діяльності спеціалізоване статистичне програмне забезпечення.

РН-17. Знати методи моделювання природничих та/або соціальних процесів.

РН-21. Вміти застосовувати у професійній діяльності знання та навички з машинного навчання, обробки зображень і природної

	мови, інших галузей науки про дані.				
Ключові слова	алгоритм, програма, тип даних, IDLE, інструкція, функція, об'єкт, клас, поліморфізм, наслідування, інкапсуляція, Python.				
Формат курсу	Очний				
Теми	Тижень	Тема, план, короткі тези	Форма діяльності	Завдання, год	Термін виконання
	<i>Семестр 1</i>				
	1	Вступ до дисципліни. <i>Предмет і завдання інформатики. Класична структура ЕОМ і ПК. Класифікація програмного забезпечення. Системи числення. Кодування символів. Представлен-ня чисел у пам'яті комп'ютера. Алгоритм. Блок-схема. Програма. Мови програмування.</i>	лекція	2	
		<i>Системи числення</i>	лабораторне заняття	2	у межах заняття
	2	Знайомство з Python. <i>Місце Python серед мов програмування. Середовища розробки. Режими виконання коду. Типи помилок. Налаштування програми. Базова термінологія. Основні типи. Основи введення-виведення.</i>	лекція	2	
		<i>IDLE. Запуск та налаштування програм.</i>	лабораторне заняття	2	у межах заняття
	3-4	Числові типи. <i>Числові типи. Операції з числами. Вбудовані функції та методи. Перетворення типів. Модулі math, decimal, fraction і random.</i>	лекція	2	
	Модулі та атрибути. <i>Стандартна бібліотека. Точкова</i>	лекція	2		

	нотація. Способи імпортування модулів. Огляд модулів стандартної бібліотеки.			
	Програмна реалізація лінійних алгоритмів	лабораторне заняття	4	у межах заняття
5-6	Галуження. Умовні інструкції. Логічний тип. Логічні оператори та вирази. Інструкції <i>if</i> і <i>match</i> .	лекція	2	
	Цикли. Функція <i>range()</i> . Цикл <i>for</i> . Цикл <i>while</i> . Інструкції <i>break</i> і <i>continue</i> .	лекція	2	
	Програмна реалізація алгоритмів з галуженнями та циклами	лабораторне заняття	4	наступне лаб. заняття
7	Рядки. Літерали рядків. Символи. Керівні символи. Індеси. Зрізи. Конкатенація і повторення рядків. Форматування рядків. Функції та методи рядків. Перевірка на входження. Посимвольний обхід рядка.	лекція	2	
	Обробка текстових даних	лабораторне заняття	2	наступне лаб. заняття
8	Файли. Запис у файл. Зчитування з файлу. Режими відкриття файлів. Інструкція <i>with</i> . Абсолютні та відносні шляхи.	лекція	2	
	Взаємодія з файлами	лабораторне заняття	2	наступне лаб. заняття
9-11	Колекції. Списки. Кортежі. Словники. Множини. Доступ до елементів колекцій. Змінюваність і незмінюваність. Накладення імен. Перевірка на	лекція	6	

	входження і поелементний обхід колекцій. Заповнення колекцій під час виконання програми. Генерація колекцій.			
	Програмування з використанням колекцій	лабораторне заняття	6	наступне лаб. заняття
12-14	Процедурне програмування. Створення і виклик функції. Інструкція <code>return</code> . Області видимості. Локальні і глобальні змінні. Способи передавання аргументів. Запаковування/розпаковування аргументів. Декомпозиція.	лекція	4	
	Розширені можливості функцій. Лямбда-функції. Оперування функціями як об'єктами. Декоратори. Основи функціонального програмування.	лекція	2	
	Процедурне програмування. Декомпозиція	лабораторне заняття	6	до кінця семестру
15	Рекурсія. Поняття рекурсії. Програмування рекурсивних функцій. Засоби підвищення ефективності рекурсивних функцій. Застосування рекурсії для розв'язування прикладних задач.	лекція	2	
	Програмна реалізація рекурсії.	лабораторне заняття	2	в межах заняття
16	Підсумкова лекція		2	
	Підсумкове заняття. Залік	лабораторне заняття	2	
2-й семестр				
1	Вступ до ООП. Об'єкти, класи та	лекція	2	

	<p>екземпляри. Успадкування. Поліморфізм. Композиція. Інкапсуляція. Інтроекція. Метапрограмування. Поняття про шаблони проектування.</p>			
	Робота з екземпляром класу на прикладі черепашичної графіки	лабораторне заняття	2	у межах заняття
2-5	<p>Основні операції над класами. Створення нових типів. Користувацькі складені типи. Атрибути і методи. Створення екземплярів класу. Ініціалізація атрибутів. Доступ до атрибутів.</p>	лекція	2	
	<p>Розширені операції над класами. Конструктори та деструктори. Вивід об'єктів на екран. Перевантаження операторів. Інкапсуляція. Управління доступом до атрибутів. Композиція</p>	лекція	2	
	<p>Наслідування. Створення нових класів з допомогою успадкування. Успадкування від базового класу. Розширення класу через успадкування. Перевизначення методів. Виклик методів базового класу. Застосування похідних класів.</p>	лекція	2	
	<p>Обробка винятків. Типи винятків. Поняття обробки винятків. Оператори TRY, EXCEPT, ELSE, FINALLY, RAISE. Блок обробки винятків. Оператор ASSERT.</p>	лекція	2	
	Об'єктно-орієнтоване програмування	лабораторне заняття	8	6 тижнів

	6-7	Прийоми ефективного програмування. <i>Ітератори, генератори, декоратори, менеджери контексту.</i>	лекція	4	
		<i>Прийоми ефективного програмування</i>	лабораторне заняття	4	у межах заняття
	8	Тестування. Модуль unittest.	лекція	2	
		<i>Модульне тестування</i>	лабораторне заняття	2	у межах заняття
	9-10	Основи програмування графічних інтерфейсів. <i>Графічний інтерфейс користувача. Модуль tkinter. Базове вікно. Елементи управління (мітки, кнопки, текстові поля і області, прапорці опцій, перемикачі). Обробка подій.</i>	лекція	4	
		<i>Програмування GUI</i>	лабораторне заняття	4	у межах заняття
	11-15	Інтерактивний комп'ютинг. Jupyter Notebook. Використання графічних елементів керування (віджетів).	лекція	4	
		Бібліотека NumPy.	лекція	2	
		Обробка і візуалізація даних. Огляд сторонніх бібліотек для статистичного аналізу та візуалізації даних. Основи роботи з бібліотеками pandas і matplotlib.	лекція	4	
		<i>Програмування в середовищі Jupyter Notebook. Обробка числових масивів. Статистичний аналіз і візуалізація даних.</i>	лабораторне заняття	10	до кінця семестру
	16	Підсумкова лекція	лекція	2	
		<i>Розбір типової практичної частини екзаменаційного</i>	лабораторне заняття	2	у межах лаб. заняття

	<table border="1"> <tr> <td></td> <td>завдання</td> <td></td> <td></td> <td></td> </tr> </table>		завдання			
	завдання					
Підсумковий контроль, форма	Залік у кінці 1-го семестру Іспит у кінці 2-го семестру					
Пререквізити	Для вивчення курсу студенти потребують базових знань з шкільного курсу інформатики. Для розуміння деяких алгоритмів потрібні базові знання з окремих тем дисциплін "Лінійна алгебра і аналітична геометрія", "Дискретна математика" і "Дискретні моделі статистики", які вивчаються одночасно з нею.					
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	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 середовищі Microsoft Teams.					
Необхідне обладнання	Для проведення лекцій: комп'ютер, проектор, доступ до мережі Інтернет. Для проведення лабораторних та виконання завдань: комп'ютер, ОС Windows/Linux, доступ до мережі Інтернет; прикладне програмне забезпечення: інтерпретатор Python 3.x, Python IDLE.					
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться за 100-бальною шкалою. <i>1-й семестр</i> Розподіл балів: <ul style="list-style-type: none"> – виконання та захист 7-ми лабораторних робіт – 70 балів; – активність на аудиторних заняттях – 15 балів; – підсумковий тест – 15 балів. <i>2-й семестр</i> Розподіл балів: <ul style="list-style-type: none"> – виконання та захист програмних проєктів у межах лабораторних занять – 50 балів (2 проєкти – 30 і 20 балів відповідно); – виконання екзаменаційного завдання – 50 балів (теоретична (тестова) частина – 25 балів, практична частина (написання і виконання коду) – 25 балів). <p>Кожна лабораторна робота чи програмний проєкт має свої максимальну оцінку і критерії оцінювання (надаються студентові разом з умовою завдання), які враховують:</p> <ul style="list-style-type: none"> – частку роботи, виконану в межах лабораторного заняття; – відповідність результатів поставленим завданням; – оформлення коду згідно з PEP 8; – представлення і захист роботи. <p>Бали за активність на аудиторних заняттях нараховуються за:</p> <ul style="list-style-type: none"> – відповідь на запитання викладача чи іншого студента; – виявлення помилки у коді, який написаний іншим студентом; 					

	<ul style="list-style-type: none"> – альтернативний спосіб вирішення завдання, яке розглядається на занятті; – змістовне запитання за темою заняття. <p>У екзаменаційному завданні прописані кількість балів за правильну відповідь на кожне питання теоретичної частини і правильну програмну реалізацію кожної структурного елемента завдання практичної частини (із типовим варіантом екзаменаційного білету студенти ознайомлюються на першому занятті другого семестру, а його розбір здійснюється на останньому лабораторному занятті і, за потреби, на консультації до іспиту).</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.</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>Критерії оцінювання результатів неформальної освіти: Студент має право отримати бали поточного контролю за участь у наукових конференціях, публікацію тез доповідей чи наукової статті, участь у конкурсах, отримання сертифікатів на освітніх платформах (Coursera, Prometheus тощо) чи курсах провідних ІТ-компаній за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни.</p>
<p>Типові питання</p>	<p>Тематика теоретичної (тестової) частини:</p> <ol style="list-style-type: none"> 1. Базові типи об'єктів Python. 2. Особливості та наслідки динамічної типізації. 3. Операції над числовими типами. 4. Операції над рядками. 5. Упорядковані колекції об'єктів. 6. Невпорядковані колекції об'єктів. 7. Умовні інструкції та цикли. 8. Основи процедурного програмування. 9. Області видимості. 10. Спеціальні режими співставлення аргументів. 11. Анонімні функції. 12. Оперування функціями як об'єктами. 13. Засоби функціонального програмування в Python. 14. Модулі та пакети.

	<p>15. Класи і ООП. 16. Перевантаження операторів. 17. Наслідування. 18. Ітератори та генератори. 19. Обробка винятків. 20. Декоратори. 21. Первинний статистичний аналіз і візуалізація даних</p> <p>Для успішного складання практичної частини (написання і виконання коду) потрібно вміти:</p> <ol style="list-style-type: none"> 1. Використовувати інструкції галуження та циклів, зокрема, з інструкціями break, continue і блоком else в циклах. 2. Використовувати стандартні контейнери: кортежі, списки, словники, множини. 3. Оголошувати та використовувати функції з позиційними та іменованими аргументами, параметрами зі значеннями за замовчуванням, у тому числі і зі змінною кількістю аргументів. 4. Оперувати функціями як об'єктами. 5. Використовувати лямбда-функції. 6. Оголошувати та використовувати декоратори функцій. 7. Оголошувати класи, визначаючи конструктори, методи, статичні атрибути, перевантажуючи оператори. 8. Наслідувати класи, перевизначати методи. 9. Створювати ітератори та генератори. 10. Оголошувати, запускати та опрацьовувати винятки, розпізнаючи тип винятку та гарантуючи виконання певних дій незалежно від виникнення винятків. 11. Взаємодіяти з файловою системою комп'ютера. 12. Будувати основні типи графіків засобами бібліотеки matplotlib, налаштовувати їх параметри. 13. Використовувати засоби бібліотеки numpy для опрацювання числових масивів. 14. Створювати jupyter-ноутбуки з розміткою для форматування тексту та набору математичних формул, записувати та виконувати фрагменти програм, використовувати графічні елементи керування (widgets) для створення інтерактивного інтерфейсу користувача. 15. Використовувати засоби бібліотеки pandas для опрацювання таблиць гетерогенних даних (завантаження таблиць, об'єднання таблиць, заповнення пропусків, вибірка, фільтрування, використання методів агрегації).
<p>Опитування</p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.</p>