

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 31 серпня 2022 р.)



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

Силабус навчальної дисципліни
«Програмування» (III семестр),
що викладається в межах ОПП Кібербезпека
першого (бакалаврського) рівня вищої освіти для здобувачів з
спеціальності 125 – Кібербезпека

Назва дисципліни	Програмування. Частина 2
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	12 – інформаційні технології 125 – кібербезпека
Викладачі дисципліни	Музичук Анатолій Омелянович, к. ф.-м. н., доц., доцент кафедри програмування Селіверстов Роман Григорович, к. ф.-м. н., доц., доцент кафедри програмування
Контактна інформація викладачів	<i>Електронна пошта:</i> anatoliy.muzychuk@lnu.edu.ua roman.seliverstov@lnu.edu.ua <i>Вебсторінки:</i> https://ami.lnu.edu.ua/employee/muzychuk 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/prohramuvannia-chastyna-2-python
Інформація про дисципліну	Курс "Програмування" (друга частина) є нормативною навчальною дисципліною зі спеціальності 125 – кібербезпека для освітньої програми Кібербезпека, яку викладають у третьому семестрі в обсязі 4 кредитів (за Європейською кредитно-трансферною системою ECTS)
Коротка анотація дисципліни	Курс фокусується на особливостях моделі даних мови Python та на бібліотечних засобах їхнього аналізу та візуалізації.
Мета та цілі дисципліни	Мета – формування компетенцій, необхідних для ефективного використання бібліотек мови Python для вирішення прикладних 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>	<p><i>Основна:</i></p> <ol style="list-style-type: none"> 1. Маттес Е. Пришвидшений курс Python. – Львів : ВСЛ, 2021. 2. Селіверстов Р., Мельничин А. Основи програмування мовою Python: навч. посібник. – Львів : ЛНУ імені Івана Франка, 2020. 3. The Python Tutorial. – https://docs.python.org/3/tutorial/index.html 4. Lutz M. Learning Python, 5th Edition. – O'Reilly Media, 2013. 5. Lambert K. A. Fundamentals of Python: First Programs, 2nd Edition. – Cengage, 2019. 6. NumPy. – http://numpy.org . 7. Pandas. – http://pandas.pydata.org . 8. Matplotlib. – http://matplotlib.org . <p><i>Додаткова:</i></p> <ol style="list-style-type: none"> 9. Wentworth P., Elkner J., Downey A., Meyers C. How to Think Like a Computer Scientist: Learning with Python 3. – Green Tea Press, 2018. 10. Python Tricks: The Book. — Dan Bader, 2017. 11. Sweigart A. Automate the Boring Stuff with Python: Practical Programming for Total Beginners. – No Starch Press, 2014. 12. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 2nd Edition. – O'Reilly Media, 2018.
<p>Обсяг курсу</p>	<p>4 кредити ЄКТС (120 годин). З них 32 години лекцій, 32 години лабораторних занять, 56 годин самостійної роботи.</p>
<p>Очікувані результати навчання</p>	<p>Після завершення цього курсу студент буде <i>знати:</i></p> <ul style="list-style-type: none"> – синтаксис, основні конструкції та типи даних 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 Python. <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;

	<ul style="list-style-type: none"> - розробляти графічні інтерфейси; - представляти, структурувати, обробляти та візуалізувати дані у комп'ютерних системах; - застосовувати одержані знання для розв'язання конкретних професійних задач. 																							
Компетентності	Інтегральна: КІ Загальні (ЗК): КЗ 1, КЗ 3, КЗ 5 Фахові компетентності спеціальності: КФ 1, КФ 2, КФ 7.																							
Програмні результати навчання	ПРН 1, ПРН 2, ПРН 5, ПРН 10, ПРН 53																							
Ключові слова	алгоритм, скрипт, тип даних, інструкція, функція, об'єкт, клас, поліморфізм, наслідування, інкапсуляція																							
Формат курсу	Очний: проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету (в умовах карантину – онлайнвий на платформі Microsoft Teams)																							
Теми	<table border="1"> <thead> <tr> <th>Тижде нь</th> <th>Тема, план, короткі тези</th> <th>Форма діяльності</th> <th>Тривалість, год</th> <th>Терм викона</th> </tr> </thead> <tbody> <tr> <td rowspan="2">1</td> <td><i>Модель даних, базові типи, оператори та інструкції Python: Режими виконання коду. Основні прийоми роботи в Jupyter Notebook. Коментування коду, довідка Python. Особливості числових типів. Введення-виведення даних. Стандартна бібліотека Python. Імпорт.</i></td> <td><i>лекція</i></td> <td>2</td> <td></td> </tr> <tr> <td><i>Програмна реалізація лінійних алгоритмів</i></td> <td><i>лабораторне заняття</i></td> <td>2</td> <td><i>наступ лаб. занятт</i></td> </tr> <tr> <td rowspan="2">2</td> <td><i>Розгалуження та цикли. Розгалуження в алгоритмах. Логічні вирази. Умовні конструкції if. Функція range(). Цикли for і while. Операції break і continue. Блок else. Визначення функцій та простих типів</i></td> <td><i>лекція</i></td> <td>2</td> <td></td> </tr> <tr> <td><i>Умовні конструкції та цикли</i></td> <td><i>лабораторне заняття</i></td> <td>2</td> <td><i>наступ лаб. занятт</i></td> </tr> </tbody> </table>	Тижде нь	Тема, план, короткі тези	Форма діяльності	Тривалість, год	Терм викона	1	<i>Модель даних, базові типи, оператори та інструкції Python: Режими виконання коду. Основні прийоми роботи в Jupyter Notebook. Коментування коду, довідка Python. Особливості числових типів. Введення-виведення даних. Стандартна бібліотека Python. Імпорт.</i>	<i>лекція</i>	2		<i>Програмна реалізація лінійних алгоритмів</i>	<i>лабораторне заняття</i>	2	<i>наступ лаб. занятт</i>	2	<i>Розгалуження та цикли. Розгалуження в алгоритмах. Логічні вирази. Умовні конструкції if. Функція range(). Цикли for і while. Операції break і continue. Блок else. Визначення функцій та простих типів</i>	<i>лекція</i>	2		<i>Умовні конструкції та цикли</i>	<i>лабораторне заняття</i>	2	<i>наступ лаб. занятт</i>
	Тижде нь	Тема, план, короткі тези	Форма діяльності	Тривалість, год	Терм викона																			
	1	<i>Модель даних, базові типи, оператори та інструкції Python: Режими виконання коду. Основні прийоми роботи в Jupyter Notebook. Коментування коду, довідка Python. Особливості числових типів. Введення-виведення даних. Стандартна бібліотека Python. Імпорт.</i>	<i>лекція</i>	2																				
		<i>Програмна реалізація лінійних алгоритмів</i>	<i>лабораторне заняття</i>	2	<i>наступ лаб. занятт</i>																			
	2	<i>Розгалуження та цикли. Розгалуження в алгоритмах. Логічні вирази. Умовні конструкції if. Функція range(). Цикли for і while. Операції break і continue. Блок else. Визначення функцій та простих типів</i>	<i>лекція</i>	2																				
<i>Умовні конструкції та цикли</i>		<i>лабораторне заняття</i>	2	<i>наступ лаб. занятт</i>																				

	3	Рядки та файли: 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 <i>with/as</i> . Шляхи.	лекція	2		
		Програмування з використанням рядків і файлів	лабораторне заняття	2	наступне лаб. заняття	
	4	Колекції: Списки. Кортежі. Словники. Множини. Доступ до елементів. Перевірка на входження та поелементний обхід. Основні функції та методи. Генерування колекцій.	лекція	2		
		Програмування з використанням колекцій	лабораторне заняття	2	наступне лаб. заняття	
	5	Функції: Декомпозиція. Функції та аргументи. Створення та виклик функції. Інструкція <i>return</i> . "Мертвий" код. Області видимості. Локальні та глобальні змінні. Вкладення функцій. Лямбда-функції. Позиційні та іменовані аргументи. Значення за замовчуванням. Передавання довільної кількості аргументів. Розпаковування аргументів. Спеціальні режими зіставлення аргументів. Рекурсивні функції. Поняття про обробку даних засобами <i>numpy</i> та візуалізація за допомогою <i>matplotlib</i>	лекція	2		

	Процедурне програмування. Декомпозиція	лабораторне заняття	2	наступне лаб. заняття	
6-7	Класи та об'єкти (екземпляри). Атрибути та методи, особливості інкапсуляції. Створення екземплярів класу. Методи <code>__init__()</code> та <code>__str__()</code> . Особливості керування доступом до атрибутів; <code>property</code> . Змінні та методи класу. Статичні методи. Перевантаження операторів. Наслідування класів, особливості множинного наслідування. Перевизначення методів. Виклик методів базового класу.	лекція	4		
	Об'єктно-орієнтоване програмування	лабораторне заняття	8	4 тижні	
8	Функція як об'єкт, атрибути функції. Декоратори функцій та класів.	лекція	2		
9	Обробка винятків. Ієрархія класів винятків. Менеджер контексту <code>with/as</code> . Протокол керування контекстом.	лекція	2		
10	Прийоми ефективного використання вбудованих колекцій; ітератори, генератори. Інструкція <code>assert</code> . Модульне тестування: <code>unittest</code> .	лекція	2		
	Тестування програм	лабораторне заняття	2	наступне лаб. заняття	
11	Основи програмування графічних інтерфейсів. Графічний інтерфейс користувача. Модуль <code>tkinter</code> . Базове вікно. Елементи управління (мітки, кнопки, текстові поля і області, прапорці	лекція	2		

		опцій, перемикачі). Обробка подій.				
		Програмування графічних інтерфейсів	лабораторне заняття	2	наступне лаб. заняття	
	12	Візуалізація даних засобами matplotlib.	лекція	2		
		Візуалізація даних з matplotlib.	лабораторне заняття	2	наступне лаб. заняття	
	13	Інтерактивний комп'ютинг: Jupyter Notebook.	лекція	2		
		Робота в середовищі Jupyter Notebook.	лабораторне заняття	2	наступне лаб. заняття	
	14	Матрично-векторні обчислення засобами NumPy.	лекція	2		
		Операції над матрицями в NumPy.	лабораторне заняття	2	наступне лаб. заняття	
	15	Аналіз гетерогенних даних засобами Pandas.	лекція	2		
		Статистичний аналіз даних	лабораторне заняття	2	наступне лаб. заняття	
	16	Імпорт C++ коду. Підсумки	лекція	2		
		Підсумкове заняття	лабораторне заняття	2		
Підсумковий контроль, форма	іспит у кінці семестру					
Пререквізити	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.					
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 середовищах Microsoft Teams, Slack, Skype тощо.					
Необхідне обладнання	Для проведення лекцій: комп'ютер, проєктор, доступ до мережі Інтернет. Для проведення лабораторних занять та виконання завдань: комп'ютер, ОС Windows/Linux, доступ до інтернету.					

<p>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</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 не менше 10 лабораторних робіт, кожна 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).</p> <p>Студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів робіт, 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 їхніми 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>Питання до екзамену</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. 15. Класи і ООП. 16. Перевантаження операторів. 17. Наслідування. 18. Ітератори та генератори. 19. Обробка винятків. 20. Декоратори. 21. Засоби візуалізації даних matplotlib 22. Багатовимірні масиви NumPy 23. Засоби аналізу даних Pandas

Для успішного складання практичної частини (програмна реалізація) потрібно вміти:

1. Використовувати інструкції галуження та циклів, зокрема, з інструкціями `break`, `continue` і блоком `else` в циклах.
2. Використовувати стандартні контейнери: рядки, 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: оголошувати функції всередині інших функцій, передавати функції аргументами інших функцій, додавати атрибути функціям тощо.
5. Використовувати лямбда-функції.
6. Оголошувати та використовувати декоратори функцій, у тому числі декоратори з параметрами.
7. Оголошувати класи, визначаючи конструктори, методи, статичні атрибути, перевантажуючи оператори.
8. Створювати ітератори та генератори.
9. Наслідувати класи, перевизначати методи.
10. Оголошувати, запускати та опрацьовувати винятки, розпізнаючи тип винятку, отримавши доступ до його даних та гарантуючи виконання певних дій незалежно від виникнення винятків.
11. Взаємодіяти з файловою системою комп'ютера: знаходити, створювати, переіменувати, вилучати файли та каталоги.
12. Завантажувати дані з файлів (текстових і двійкових), серіалізувати дані.
13. Будувати основні типи графіків засобами бібліотеки `matplotlib`, налаштовувати їх параметри.
14. Використовувати засоби бібліотеки `numpy` для опрацювання числових масивів.
15. 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.
16. Використовувати засоби бібліотеки `pandas` для опрацювання таблиць гетерогенних даних (завантаження таблиць, об'єднання таблиць, заповнення пропусків, вибірка, фільтрування, використання методів агрегації, застосування функцій користувача).

Опитування

Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.