

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
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
«Вступ до програмування»,
що викладається в межах першого (бакалаврського) рівня
вищої освіти для здобувачів
з спеціальності 113 Прикладна математика

Львів 2023 р.

Назва дисципліни	Вступ до програмування
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: 113 Прикладна математика
Викладачі дисципліни	Черняхівський Володимир Вікторович, к. ф.-м. н., доц., доцент кафедри програмування
Контактна інформація викладачів	Електронна пошта: volodymyr.chernyakhivskyy@lnu.edu.ua веб-сторінка: https://ami.lnu.edu.ua/employee/cherniakhivskyy
Консультації з питань навчання по дисципліні відбуваються	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі онлайн консультації через Zoom чи Microsoft Teams. Для погодження часу онлайн консультацій потрібно писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/course/introduction-to-programming-system-analysis
Інформація про дисципліну	Курс «Вступ до програмування» є нормативною дисципліною зі спеціальності 113 Прикладна математика, яку викладають у першому семестрі в обсязі 5 кредитів (за Європейською кредитно-трансферною системою ECTS).
Коротка анотація дисципліни	Курс є базовим для підготовки студентів спеціальності «Прикладна математика». Викладання курсу має на меті сформувати у студентів базову систему знань та навиків в області сучасних прийомів прикладного програмування та застосування комп'ютерів. Розглядають різні аспекти розробки програм: постановка задачі; аналіз постановки задачі; вибір методу розв'язування задачі; розробка алгоритму; оцінка вхідних і вихідних даних; аналіз правильності алгоритму; складання програми алгоритмічною мовою Python; компіляція і виправлення синтаксичних помилок; складання тестів для перевірки правильності програми; прийоми роботи в обраній системі програмування для мови Python; робота з файловою системою операційної системи.
Мета та цілі дисципліни	Метою нормативної дисципліни «Вступ до програмування» є: <ul style="list-style-type: none"> • формування системи базових знань про прийоми проектування алгоритмів і програмування типових базових задач за різними розділами прикладних областей; • вивчення мови Python, можливостей її застосування, загальних принципів розв'язування задач за допомогою програмування алгоритмічною мовою Python; • вивчення типових сучасних прийомів проектування і програмування базових задач різного ступеня складності, отримання знань про будову інформаційних моделей для програмування; • формування практичних навичок роботи на комп'ютері щодо процесу розробки програм на основі алгоритмічної мови Python;
Література для вивчення дисципліни	<i>Основна література</i> <ol style="list-style-type: none"> 1. Python Software Foundation. The Python Tutorial [Електронний ресурс]. – Режим доступу: https://docs.python.org/3/tutorial/index.html 2. Python Software Foundation. Python 3.10.0 documentation [Електронний ресурс]. – Режим доступу: https://docs.python.org/3.10/ 3. Lutz M. Learning Python, 5th Edition / M.Lutz // O'Reilly Media, Inc. – 2013. – 1648 p. 4. Маттес Ерік. Пришвидшений курс Python. Практичний, проектно-орієнтований вступ до програмування: Пер. з англ. / Ерік Маттес // Видавництво Старого Лева. – 2021. – 600с.

	<p>5. Беррі Пол. Head First. Python: Пер. з англ. / Пол Беррі // Фабула. – 2021. – 624с.</p> <p>6. Васильєв Олексій. Програмування мовою Python / Олексій Васильєв // Навчальна книга – Богдан. – 2019. – 504с.</p> <p>7. Селіверстов Р., Мельничин А. Основи програмування мовою Python: навч. посібник. – Львів : ЛНУ імені Івана Франка. – 2020.</p> <p>8. W3Schools Online Web Tutorials. Python Tutorial [Електронний ресурс]. – Режим доступу: https://www.w3schools.com/python/default.asp</p> <p>9. Lambert K. A. Fundamentals of Python: First Programs, 2nd Edition. – Cengage, 2019. – 498 p.</p> <p>10. Вікіпідручник. Підручник мови Python [Електронний ресурс]. – Режим доступу: https://uk.wikibooks.org/wiki/Підручник_мови_Python</p> <p><i>Додаткова література</i></p> <p>11. Codecademy. Learn Python 3 [Електронний ресурс]. – Режим доступу: https://www.codecademy.com/learn/learn-python-3</p> <p>12. Список структур даних. Матеріал з Вікіпедії – вільної енциклопедії. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Список_структур_даних</p>
Обсяг курсу	5 кредитів ЄКТС – 150 годин. З них 32 годин лекцій, 32 годин лабораторних занять та 86 годин самостійної роботи.
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> ● предмет, методи та завдання дисципліни; ● математичні та інформаційні основи програмування, як методу розв'язування задач; ● алгоритмічну мову Python в обсязі навчальної програми; ● типові сучасні алгоритми програмування базових прикладних задач різного тематичного змісту; ● процедури проектування, програмування, тестування і налагодження програм мовою Python; <p><i>вміти</i></p> <ul style="list-style-type: none"> ● проектувати алгоритми розв'язування задач; ● скласти, тестувати і налагоджувати програми мовою Python; ● аналізувати і модифікувати програми до зміни умов задачі; ● застосовувати стандартні функції і бібліотеки функцій; ● працювати на комп'ютері в обраному середовищі програмування мови Python. <p>Курс забезпечує набуття таких компетентностей та програмних результатів навчання:</p> <p>Фахові компетентності спеціальності:</p> <p>ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.</p> <p>Програмні результати навчання:</p> <p>РН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів.</p>
Ключові слова	Алгоритм, виконавець алгоритму, алгоритмічна мова, компіляція, інтерпретація, програма, сценарій, ідентифікатор, змінна величина, об'єкт, посилання, програмна операція, програмна функція, оператор, вираз, логічне висловлювання, логічний вираз, умовний вираз, умовний оператор, циклічний алгоритм, оператор циклу, символьний рядок, код символа, операція над рядком, лексикографічний порядок, конкатенація, індексування, зріз рядка, форматування, тип списку, конструктор,

генератор, ітерований об'єкт, тип матриці, тип словника, ключ словника, значення словника, доступ за ключем, генератор словника, об'єкт відображення словника, тип множини, конструктор множини, тип кортежа, файл, операція з файлом, визначена функція, аргумент функції, параметр функції, локальна змінна, поліморфізм функції, позиційний аргумент, іменованний аргумент, область видимості, простір імен, вкладена функція, високорівнева функція, лямбда-вираз, рекурсивна функція, модуль Python, імпортування, атрибут модуля, самотестування модуля, особлива ситуація, виключення, тип виключення, клас, об'єкт, конструктор класу, технологія ООП, метод класу, перевантаження оператора.

Формат курсу

Очний: проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах форсмажорних обставин – онлайн-овий на платформі Microsoft Teams.

Зв'язки елементів курсу:

```

graph TD
    A[Вступ до програмування] --> B[Лекції]
    A --> C[Лабораторні]
    B --> D[1 лекція на тиждень]
    C --> E[1 заняття на тиждень]
    B --> F[Екзамен: оцінка 0-50]
    C --> F
    C --> G[Сумарна оцінка 0-50]
    F --> H[Підсумкова оцінка 0-100 балів]
    G --> H
  
```

Теми

Тиж-день	Тема, план, короткі тези	Форма заняття	Тривалість год	Термін виконання
1	Задачі, алгоритми, програми. Загальна схема процесу програмування. Приклад задачі. Загальне поняття алгоритму. Алгоритм розв'язку задачі. Реалізація алгоритму алгоритмічною мовою. Компіляція, інтерпретація, виконання.	Лекція, Самостійна робота	2 5	
	Системи числення: двійкова, шістнадцяткова, вісімкова. Переведення між системами числення. Правила переведення. Цілі числа, дробові.	Лабораторна робота	2	Наступне лабораторне заняття
2	Числові типи. Оператор присвоєння. Ідентифікатори. Змінні величини, об'єкти, посилання. Принципи будови системи даних. Числа і операції. Деякі числові функції. Оператор присвоєння. Функції друкування і читання даних для консолі. Приклади програм (сценаріїв) для числових типів.	Лекція, Самостійна робота	2 6	
	1)Прямі арифметичні операції в системах числення. 2)Письмове опитування (контрольна робота) за темами лабораторних робіт 1 і 2 (системи числення).	Лабораторна робота, контрольна	2	Наступне лабораторне заняття
3	Умовні вирази. Умовний оператор. Логічний тип даних. Порівняння. Умовний вираз. Приклади на обчислення логічних виразів. Умовний оператор if. Приклади задач на умовний оператор. Перехід до стилю Python.	Лекція, Самостійна робота	2 5	
	Середовище розробки Python's IDLE. Задачі і програмна реалізація прикладів за змістом тем 2 і 3.	Лабораторна робота	2	Наступне лабораторне заняття
4	Оператори циклу. Цикли для скалярних даних. Загальна ідея циклічних алгоритмів. Оператор циклу while. Приклади програмування на застосування циклу while. Функція генерування цілих чисел. Оператор циклу for. Деякі задачі для циклу for. Повна	Лекція, Самостійна робота	2 5	

	форма циклу for. Задачі і програмна реалізація прикладів за змістом теми 4.	Лабораторна робота	2	Наступне лабораторне заняття
5	Текстові рядки і літери. Текст, зображення, кодування. Система Unicode. Функції ord(), chr(). Таблиця UTF-8. Базові операції над рядками. Лексикографічний порядок. Демонстрація окремих операцій. Індування літер рядка, зрізи рядка. Приклади задач опрацювання рядків. Форматування рядків. Задачі і програмна реалізація прикладів за змістом теми 5.	Лекція, Самостійна робота Лабораторна робота	2 5	
6	Списки. Структура списків. Прийоми створення списків. Генератори списків. Друкування списків. Типові операції для списків. Деякі особливості роботи з списками. Приклади задач опрацювання списків. Додаткові прийоми створення списків. Задачі і програмна реалізація прикладів за змістом теми 6.	Лекція, Самостійна робота Лабораторна робота	2 5	
7	Матриці. Структура матриці та її зображення в програмі. Створення матриць. Читання даних матриці. Друкування матриць. Приклади задач опрацювання матриць. 1)Задачі і програмна реалізація прикладів за змістом теми 7. 2)Контрольна робота/завдання за темами 2-7.	Лекція, Самостійна робота Лабораторна робота; контрольна	2 7	
8	Словники. Загальна характеристика словників. Доступ до елементів словника, ключі. Базові операції над словниками. Створення словників. Генератори словників. Правила доступу до елементів словника. Робота з ключами, робота зі значеннями. Друкування елементів словника. Приклади задач з використанням словників. Задачі і програмна реалізація прикладів за змістом теми 8.	Лекція, Самостійна робота Лабораторна робота	2 5	
9	Множини, кортежі. Множини як математичні об'єкти. Створення множин, конструктор, літерал, генератор. Друкування множин. Основні операції над множинами. Приклади задач з використанням множин. Кортежі, загальна характеристика. Створення кортежів, конструктор кортежа. Друкування кортежів. Основні операції над кортежами. Приклади задач з використанням кортежів. Задачі і програмна реалізація прикладів за змістом теми 9.	Лекція, Самостійна робота Лабораторна робота	2 5	
10	Файли. Загальні принципи роботи з файлами. Базові операції файлів. Читання, запис файлів. Приклади задач з використанням файлів. 1)Задачі і програмна реалізація прикладів за змістом теми 10. 2)Контрольна робота/завдання за темами 8-10.	Лекція, Самостійна робота Лабораторна робота; контрольна	2 6	
11	Функції, визначені програмою. Загальна ідея програмованих функцій. Створення функцій. Виклики і виконання функцій. Локальні змінні функцій. Функції без повернення результату. Поліморфізм функцій. Аргументи функцій, передавання змінних і незмінних об'єктів, особливості, аналіз і рекомендації. Задачі і програмна реалізація прикладів за змістом теми 11.	Лекція, Самостійна робота Лабораторна робота	2 5	
12	Методи будови функцій. Позиційні аргументи. Іменовані аргументи. Области видимості і простори імен. Доступ до просторів імен. Вкладені функції. Високорівневі функції і операції. Анонімні функції (лямбда-вирази). Елементи функціонального програмування. Рекурсивні функції. Задачі і програмна реалізація прикладів за змістом теми 12.	Лекція, Самостійна робота Лабораторна робота	2 5	
13	Модулі та їх застосування. Загальна характеристика модулів. Імпортування і атрибути. Особливості імпортування модулів. Інструкція from. Повторне завантаження модулів. Самотестування модулів. Задачі і програмна реалізація прикладів за змістом теми 13.	Лекція, Самостійна робота Лабораторна робота	2 5	

				заняття
	14	Особливі ситуації (виключення). Характеристика особливих ситуацій. Загальні принципи опрацювання виключень. Деякі типи виключень (помилки). Приклади опрацювання виключень. Збудження виключень. Інструкція assert. Об'єднана інструкція try/except/finally.	Лекція, Самостійна робота	2 7
		1)Задачі і програмна реалізація прикладів за змістом теми 14. 2)Контрольна робота/завдання за темами 11-14.	Лабораторна робота; контрольна	2 Наступне лабораторне заняття
	15	Вступ до об'єктно-орієнтованого програмування. Класи і об'єкти: постановка задачі. Визначення класів. Проектування класу і об'єктів. Конструктор класу. Деякі способи роботи з об'єктами. Методи класу, приклади реалізації. Перевантаження операторів у класах. Друкування об'єктів. Приклад застосування перевантажених операцій. Поняття про менеджери контексту.	Лекція, Самостійна робота	2 5
		Задачі і програмна реалізація прикладів за змістом теми 15.	Лабораторна робота	2 Наступне лабораторне заняття
	16	Підсумковий огляд засобів мови Python. Роль мови Python в розробці сучасних програмних проєктів. Вимоги до володіння методами практичного програмування на основі засобів і технологій Python.	Лекція, Самостійна робота	2 5
		Додаткові задачі за темами курсу. Підготовка до іспиту.	Лабораторна робота	2 Початок сесії
Поточний контроль, форми	<p>Поточний контроль знань студентів виконується під час проведення лабораторних занять протягом семестру і має на меті перевірку рівня підготовленості студента до виконання завдань за темою, запланованою на окреме кожне заняття. Крім того, поточний контроль виконують для перевірки виконання студентом індивідуальних завдань. Форми поточного контролю:</p> <ul style="list-style-type: none"> – усне індивідуальне опитування; – письмове загальне опитування у формі задач; – опитування у формі тестування; – письмова контрольна робота; – перевірка плану виконання з теми лабораторного заняття; – експертна оцінка виконання завдань і отримання достовірних результатів; – тестування результатів програмної реалізації завдань; – перевірка правильності виконання індивідуальних завдань, виданих на самостійне опрацювання; – захист студентом індивідуальних завдань. <p>Максимальна кількість балів за поточний контроль – 50.</p>			
Екзаменаційний контроль, форма	<p>Іспит в кінці семестру.</p> <p>Екзаменаційний контроль проводиться за розкладом, визначеним навчальним графіком для складання іспитів.</p> <p>На іспит студент отримує 4 задачі з різних розділів курсу «Вступ до програмування». Всі 4 задачі потрібно реалізувати на комп'ютері мовою Python і отримати працездатні програми. Для складання програм студент може обрати будь-яке середовище програмування для мови Python. Для всіх задач достатньо консольного режиму виконання.</p> <p>До кожної реалізованої задачі студент повинний самостійно скласти тестові вхідні дані (декілька варіантів) і продемонструвати виконання програми на кожному тесті.</p> <p>Шкала оцінювання на іспиті: алгоритм і програма задач А, Б, В, Г – по 10 балів кожна, самостійне складання тестів до задач – 4 бали, реалізація технології try-except-finally для однієї чи більше задач – 4 бали, застосування лямбда-функцій чи функціонального програмування – 2 бали. Разом – 50 балів.</p> <p>Максимальна кількість балів за екзаменаційний контроль – 50.</p>			
Підсумковий контроль, шкала	<p>Підсумковий контроль включає в себе результати поточного контролю знань студентів, що проводився протягом семестру, та екзаменаційний контроль.</p>			

<p>оцінок</p>	<p>Максимальна кількість балів підсумкового контролю – 100. Шкала оцінювання підсумкового контролю:</p> <table border="1" data-bbox="475 185 1211 620"> <thead> <tr> <th>Бали</th> <th>ECTS</th> <th>Національ на шкала</th> <th>Коментар</th> </tr> </thead> <tbody> <tr> <td>90 - 100</td> <td>A</td> <td>5</td> <td>відмінно</td> </tr> <tr> <td>81 - 89</td> <td>B</td> <td>4</td> <td>дуже добре</td> </tr> <tr> <td>71 - 80</td> <td>C</td> <td>4</td> <td>добре</td> </tr> <tr> <td>61 - 70</td> <td>D</td> <td>3</td> <td>задовільно</td> </tr> <tr> <td>51 - 60</td> <td>E</td> <td>3</td> <td>достатньо</td> </tr> <tr> <td>0 - 50</td> <td>FX (F)</td> <td>2</td> <td>незадовільно</td> </tr> </tbody> </table>	Бали	ECTS	Національ на шкала	Коментар	90 - 100	A	5	відмінно	81 - 89	B	4	дуже добре	71 - 80	C	4	добре	61 - 70	D	3	задовільно	51 - 60	E	3	достатньо	0 - 50	FX (F)	2	незадовільно
Бали	ECTS	Національ на шкала	Коментар																										
90 - 100	A	5	відмінно																										
81 - 89	B	4	дуже добре																										
71 - 80	C	4	добре																										
61 - 70	D	3	задовільно																										
51 - 60	E	3	достатньо																										
0 - 50	FX (F)	2	незадовільно																										
<p>Пререквізити</p>	<p>Для вивчення курсу студенти потребують базових знань з дисциплін «Математика», «Інформатика» за курс середньої школи.</p>																												
<p>Навчальні методи та техніки, які використовують під час викладання курсу</p>	<p>Лекції з мультимедійними презентаціями та з демонстрацією прийомів програмування засобами мови Python; лабораторні заняття у вигляді проектування алгоритмів і програм, виконання практичних завдань та проєктів; індивідуальні домашні завдання на програмування розв'язків задач; самостійне опрацювання навчальних матеріалів: підручників, конспектів лекцій, електронних ресурсів, готових програм мовою Python, додаткових матеріалів, розміщених у хмарному сховищі (Microsoft Teams, Google Диск). Обговорення теоретичного та практичного матеріалу в онлайн сервісах, формулювання творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного і практичного матеріалу.</p>																												
<p>Необхідне обладнання, засоби, матеріали</p>	<p>Для проведення лекцій: комп'ютер, проектор, доступ до мережі інтернет. Для проведення лабораторних та виконання завдань: комп'ютер; ОС Windows/Linux; доступ до інтернету; середовище програмування мовою Python (IDLE тощо). Кожен студент повинен мати свій доступ до корпоративного середовища Microsoft 365 через надану персональну поштову адресу *@lnu.edu.ua, а також свій акаунт (поштову адресу) в Google *@gmail.com. Уся література і робочі матеріали, які студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p>																												
<p>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</p>	<p>Оцінювання проводиться за 100-бальною шкалою. 50 балів нараховують за виконання лабораторних завдань, домашніх завдань, контрольних робіт впродовж семестру, ще 50 балів – за виконання екзаменаційного завдання. Лабораторні завдання індивідуальні. Домашні завдання можуть бути індивідуальні та командні. Упродовж семестру студент виконує не менше 10 лабораторних і домашніх, які оцінюють різною шкалою залежно від складності (критерії надаються студентам до кожного лабораторного чи домашнього завдання). На екзамен студент отримує чотири задачі за різними розділами курсу, які треба реалізувати програмно. Перелік окремих вимог, шкала оцінювання і зразки завдань будуть надані студентам до закінчення лекційних і лабораторних занять. Оцінка за екзаменаційне завдання може отримати додаткові бали, якщо завдання чи проєкти студента, виконані впродовж семестру, мали закінчений характер і високий рівень фахової реалізації. Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Активність під час</p>																												

	<p>проведення лекцій і лабораторних заохочується балами при оцінюванні відповідного лабораторного завдання чи проєкта. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів робіт, передбачених курсом. Виконані роботи завантажують у відповідне хмарне сховище. Альтернативою відвідування лабораторних занять в університеті може бути дистанційна онлайн робота за розкладом проведення занять.</p> <p>Академічна доброчесність: очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп'ютерних програм як своїх становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p>
Опитування	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.</p>