

**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 2024 р.)



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

**Силабус навчальної дисципліни**  
**«Вступ до програмування»,**  
**що викладається в межах ОПП “Прикладна математика”**  
**для здобувачів першого (бакалаврського) рівня вищої освіти**  
**з предметної спеціальності 113 Прикладна математика**  
**галузі знань 11 Математика та статистика**

Львів 2024 р.

<b>Назва дисципліни</b>	Вступ до 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: 11 Математика та статистика Спеціальність: 113 Прикладна математика
<b>Викладачі дисципліни</b>	Черняхівський Володимир Вікторович, к. ф.-м. н., доц., доцент кафедри програмування
<b>Контактна інформація викладачів</b>	Електронна пошта: <a href="mailto:volodymyr.chernyakhivskyy@lnu.edu.ua">volodymyr.chernyakhivskyy@lnu.edu.ua</a> веб-сторінка: <a href="https://ami.lnu.edu.ua/employee/cherniakhivskyy">https://ami.lnu.edu.ua/employee/cherniakhivskyy</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	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.
<b>Сторінка курсу</b>	<a href="https://ami.lnu.edu.ua/course/introduction-to-programming-system-analysis">https://ami.lnu.edu.ua/course/introduction-to-programming-system-analysis</a>
<b>Інформація про дисципліну</b>	Курс “Вступ до програмування” є нормативною дисципліною зі спеціальності 113 Прикладна математика для освітньої програми «Прикладна математика», яку викладають у першому семестрі в обсязі 5 кредитів (за Європейською кредитно-трансферною системою ECTS).
<b>Коротка анотація дисципліни</b>	Курс є базовим для підготовки студентів факультету прикладної математики та інформатики спеціальності “Прикладна математика” в межах освітньо-професійної програми "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.
<b>Мета та цілі дисципліни</b>	Метою нормативної дисципліни «Вступ до програмування» є: <ul style="list-style-type: none"> <li>• формування системи базових знань про прийоми проектування алгоритмів і програмування типових базових задач за різними розділами прикладних областей;</li> <li>• вивчення мови Python, можливостей її застосування, загальних принципів розв'язування задач за допомогою програмування алгоритмічною мовою Python;</li> <li>• вивчення типових сучасних прийомів проектування і програмування базових задач різного ступеня складності, отримання знань про будову інформаційних моделей для програмування;</li> <li>• формування практичних навичок роботи на комп'ютері щодо процесу розробки програм на основі алгоритмічної мови Python;</li> </ul>
<b>Література для вивчення дисципліни</b>	<i>Основна література</i> <ol style="list-style-type: none"> <li>1. Python Software Foundation. The Python Tutorial [Електронний ресурс]. – Режим доступу: <a href="https://docs.python.org/3/tutorial/index.html">https://docs.python.org/3/tutorial/index.html</a></li> <li>2. Python Software Foundation. Python 3.10.0 documentation [Електронний ресурс]. – Режим доступу: <a href="https://docs.python.org/3.10/">https://docs.python.org/3.10/</a></li> <li>3. Lutz M. Learning Python, 5th Edition / M.Lutz // O'Reilly Media, Inc. – 2013. – 1648 p.</li> </ol>

	<p>4. Маттес Ерік. Пришвидшений курс Python. Практичний, проєктно-орієнтований вступ до програмування: Пер. з англ. / Ерік Маттес // Видавництво Старого Лева. – 2021. – 600с.</p> <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: <a href="https://www.w3schools.com/python/default.asp">https://www.w3schools.com/python/default.asp</a></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: <a href="https://uk.wikibooks.org/wiki/Підручник_мови_Python">https://uk.wikibooks.org/wiki/Підручник_мови_Python</a></p> <p><i>Додаткова література</i></p> <p>11. Codecademy. Learn Python 3 [Електронний ресурс]. – Режим доступу: <a href="https://www.codecademy.com/learn/learn-python-3">https://www.codecademy.com/learn/learn-python-3</a></p> <p>12. Список структур даних. Матеріал з Вікіпедії – вільної енциклопедії. [Електронний ресурс]. – Режим доступу: <a href="https://uk.wikipedia.org/wiki/Список_структур_даних">https://uk.wikipedia.org/wiki/Список_структур_даних</a></p>
<b>Обсяг курсу</b>	5 кредитів ЄКТС – 150 годин. З них 32 годин лекцій, 32 годин лабораторних занять та 86 годин самостійної роботи.
<b>Очікувані результати навчання</b>	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> <li>● предмет, методи та завдання дисципліни;</li> <li>● математичні та інформаційні основи програмування, як методу розв’язування задач;</li> <li>● алгоритмічну мову Python в обсязі навчальної програми;</li> <li>● типові сучасні алгоритми програмування базових прикладних задач різного тематичного змісту;</li> <li>● процедури проектування, програмування, тестування і налагодження програм мовою Python;</li> </ul> <p><i>вміти</i></p> <ul style="list-style-type: none"> <li>● проектувати алгоритми розв’язування задач;</li> <li>● складати, тестувати і налагоджувати програми мовою Python;</li> <li>● аналізувати і модифікувати програми до зміни умов задачі;</li> <li>● застосовувати стандартні функції і бібліотеки функцій;</li> <li>● працювати на комп’ютері в обраному середовищі програмування мови Python.</li> </ul>
<b>Компетентності</b>	<p><i>Інтегральна:</i> Здатність застосовувати загальні принципи програмування практичних задач засобами мови Python на професійному рівні. Здатність застосовувати сучасні технології програмування задач різного призначення, проектувати структури даних і алгоритмічні моделі на основі засобів мови Python.</p> <p><i>Загальні (ЗК):</i></p> <ul style="list-style-type: none"> <li>● ЗК01. Здатність учитися і оволодівати сучасними знаннями.</li> <li>● ЗК06. Здатність до абстрактного мислення, аналізу та синтезу.</li> <li>● ЗК08. Знання та розуміння предметної області та розуміння професійної діяльності.</li> </ul> <p><i>Спеціальні (фахові, предметні) компетентності (ФК):</i></p> <ul style="list-style-type: none"> <li>● ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.</li> <li>● ФК21. Здатність аналізувати можливості наявних технологій розробки програмного забезпечення, обирати та застосовувати найбільш ефективний інструментарій відповідно до специфіки завдання.</li> </ul>

<b>Програмні результати навчання</b>	<ul style="list-style-type: none"> <li>РН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів.</li> <li>РН14. Виявляти здатність до самонавчання та продовження професійного розвитку.</li> </ul>
--------------------------------------	--

<b>Ключові слова</b>	<p>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.</p>
----------------------	--

<b>Формат курсу</b>	<p>Очний: проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах форсмажорних обставин – онлайн-овий на платформі Microsoft Teams.</p> <p>Зв'язки елементів курсу:</p> <pre> graph TD     A[Вступ до програмування] --&gt; B[Лекції]     A --&gt; C[Лабораторні]     B --&gt; D[1 лекція на тиждень]     B --&gt; E[Екзамен: оцінка 0-50]     C --&gt; F[1 заняття на тиждень]     C --&gt; G[Сумарна оцінка 0-50]     E --&gt; H[Підсумкова оцінка 0-100 балів]     G --&gt; H   </pre>
---------------------	--

<b>Теми</b>	<table border="1"> <thead> <tr> <th>Тиж-день</th> <th>Тема, план, короткі тези</th> <th>Форма заняття</th> <th>Тривалість год</th> <th>Термін виконання</th> </tr> </thead> <tbody> <tr> <td rowspan="2">1</td> <td>Задачі, алгоритми, програми. Загальна схема процесу програмування. Приклад задачі. Загальне поняття алгоритму. Алгоритм розв'язку задачі. Реалізація алгоритму алгоритмічною мовою. Компіляція, інтерпретація, виконання.</td> <td>Лекція</td> <td>2</td> <td></td> </tr> <tr> <td>Системи числення: двійкова, шістнадцяткова, вісімкова. Переведення між системами числення. Правила переведення. Цілі числа, дробові.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td>2</td> <td>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</td> <td>Лекція</td> <td>2</td> <td></td> </tr> </tbody> </table>	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	2		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	Лекція	2	
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	2																	
	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	Лекція	2																	

	читання даних для консолі. Приклади програм (сценаріїв) для числових типів.			
	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	2	
	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 форма циклу for.	Лекція	2	
	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. Індексуювання літер рядка, зрізи рядка. Приклади задач опрацювання рядків. Форматування рядків.	Лекція	2	
	Задачі і програмна реалізація прикладів за змістом теми 5.	Лабораторна робота	2	Наступне лабораторне заняття
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.	Лекція	2	
	Задачі і програмна реалізація прикладів за змістом теми 6.	Лабораторна робота	2	Наступне лабораторне заняття
7	Матриці. Структура матриці та її зображення в програмі. Створення матриць. Читання даних матриці. Друкування матриць. Приклади задач опрацювання матриць.	Лекція	2	
	1)Задачі і програмна реалізація прикладів за змістом теми 7. 2)Контрольна робота/завдання за темами 2-7.	Лабораторна робота; контрольна	2	Наступне лабораторне заняття
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.	Лекція	2	
	Задачі і програмна реалізація прикладів за змістом теми 8.	Лабораторна робота	2	Наступне лабораторне заняття
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.	Лекція	2	
	Задачі і програмна реалізація прикладів за змістом теми 9.	Лабораторна робота	2	Наступне лабораторне заняття
10	Файли. Загальні принципи роботи з файлами. Базові операції файлів. Читання, запис файлів. Приклади задач з використанням файлів.	Лекція	2	
	1)Задачі і програмна реалізація прикладів за змістом теми 10. 2)Контрольна робота/завдання за темами 8-10.	Лабораторна робота; контрольна	2	Наступне лабораторне заняття
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.	Лекція	2	

		Задачі і програмна реалізація прикладів за змістом теми 11.	Лабораторна робота	2	Наступне лабораторне заняття
	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.	Лекція	2	
		Задачі і програмна реалізація прикладів за змістом теми 12.	Лабораторна робота	2	Наступне лабораторне заняття
	13	Модулі та їх застосування. Загальна характеристика модулів. Імпортування і атрибути. Особливості імпортування модулів. Інструкція from. Повторне завантаження модулів. Самотестування модулів.	Лекція	2	
		Задачі і програмна реалізація прикладів за змістом теми 13.	Лабораторна робота	2	Наступне лабораторне 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	2	
		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	2	
		Задачі і програмна реалізація прикладів за змістом теми 15.	Лабораторна робота		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	2	
		Додаткові задачі за темами курсу. Підготовка до іспиту.	Лабораторна робота		Початок сесії
<b>Поточний контроль, форми</b>	<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"> <li>– усне індивідуальне опитування;</li> <li>– письмове загальне опитування у формі задач;</li> <li>– опитування у формі тестування;</li> <li>– письмова контрольна робота;</li> <li>– перевірка плану виконання з теми лабораторного заняття;</li> <li>– експертна оцінка виконання завдань і отримання достовірних результатів;</li> <li>– тестування результатів програмної реалізації завдань;</li> <li>– перевірка правильності виконання індивідуальних завдань, виданих на самостійне опрацювання;</li> <li>– захист студентом індивідуальних завдань.</li> </ul> <p>Максимальна кількість балів за поточний контроль – 50.</p>				
<b>Екзаменаційний контроль, форма</b>	<p>Іспит в кінці семестру.  Екзаменаційний контроль проводиться за розкладом, визначеним навчальним графіком для складання іспитів.  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>Оцінка за екзаменаційне завдання може отримати додаткові бали, якщо завдання чи проекти студента, виконані впродовж семестру, мали закінчений характер і високий рівень фахової реалізації.</p> <p>Максимальна кількість балів за екзаменаційний контроль – 50.</p>																																
<p><b>Підсумковий контроль, шкала оцінок</b></p>	<p>Підсумковий контроль включає в себе результати поточного контролю знань студентів, що проводився протягом семестру, та екзаменаційний контроль.</p> <p>Максимальна кількість балів підсумкового контролю – 100.</p> <p>Шкала оцінювання підсумкового контролю:</p> <table border="1" data-bbox="491 640 1233 1133"> <thead> <tr> <th>Бали</th> <th>ECTS</th> <th>Національна 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>26 - 50</td> <td>FX</td> <td>2+</td> <td>незадовільно</td> </tr> <tr> <td>0 - 25</td> <td>F</td> <td>2-</td> <td>повторний курс</td> </tr> </tbody> </table>	Бали	ECTS	Національна 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	26 - 50	FX	2+	незадовільно	0 - 25	F	2-	повторний курс
Бали	ECTS	Національна 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																														
26 - 50	FX	2+	незадовільно																														
0 - 25	F	2-	повторний курс																														
<p><b>Пререквізити</b></p>	<p>Для вивчення курсу студенти потребують базових знань з дисциплін “Математика”, “Інформатика” за курс середньої школи. Одночасно з вивченням програмування студенти проходять навчальну комп’ютерну практику, впродовж якої виконують завдання з програмування, поглиблюють набуті в курсі знання та удосконалюють навички. Дисципліна “Вступ до програмування” також тісно пов’язана з курсом “Алгоритми обчислювальних процесів”, який вивчають у другому семестрі.</p>																																
<p><b>Навчальні методи та техніки, які використовують під час викладання курсу</b></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 проектів; 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><b>Необхідне обладнання, засоби, матеріали</b></p>	<p>Для проведення лекцій: комп’ютер, проектор, доступ до мережі інтернет.</p> <p>Для проведення лабораторних та виконання завдань: комп’ютер; ОС Windows/Linux; доступ до інтернету; середовище програмування мовою Python (IDLE тощо).</p> <p>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.</p> <p>Уся література і робочі матеріали, які студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім</p>																																

	<p>особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p>
<p><b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b></p>	<p><b>Оцінювання</b> 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). Для кожного завдання визначено термін виконання – зазвичай до наступного лабораторного заняття. Вчасно виконані завдання оцінюють так (у відсотках від максимальної оцінки):</p> <ul style="list-style-type: none"> <li>• 100% – умови завдання виконано повністю, алгоритми складено правильно, програма містить належні коментарі, роботу програми перевірено на достатньому наборі тестових даних, автор відповідає на всі запитання щодо використаних підходів, чітко інтерпретує отримані результати, немає ознак недоброчесності;</li> <li>• 80% – наведено логічно правильну послідовність розв’язування, алгоритми складено правильно, бракує окремих коментарів чи тестів, автор не досить повно пояснює використані підходи, немає ознак недоброчесності;</li> <li>• 60% – у правильній послідовності розв’язування допущено окремі помилки, які автор уміє виправити після зауваження викладача, бракує коментарів чи тестів, на запитання щодо використаних підходів автор відповідає з помилками, немає ознак недоброчесності;</li> <li>• 40% – у правильній послідовності розв’язування пропущено окремі етапи, завдання виконано частково, автор не розуміє недоліків поданої роботи, не вміє їх виправити, немає ознак недоброчесності;</li> <li>• 20% – завдання виконано частково, немає тестів, програма працює правильно для окремих наборів вхідних даних, автор не може самостійно інтерпретувати отримані результати, виправити помилки, немає ознак недоброчесності;</li> <li>• 0% – завдання не виконано, написана програма не відповідає умові, або ж виявлено ознаки недоброчесності: запозичення, фрагменти коду, дію яких автор пояснити не може, автор не володіє відповідним теоретичним матеріалом тощо;</li> <li>• можуть бути нараховані додаткові бали за повністю виконане завдання, яке містить кілька способів розв’язування, використовує особливо ефективний спосіб, демонструє креативність автора тощо.</li> </ul> <p>Запізнення терміну виконання завдання зменшує максимальну оцінку за завдання, ступінь зменшення визначає викладач до кожної роботи.</p> <p>Загальні критерії і шкала оцінювання <b>екзаменаційних завдань</b> викладені в розділі "Екзаменаційний контроль, форма". Перелік окремих вимог і зразки екзаменаційних завдань будуть надані студентам до закінчення лекційних і лабораторних занять.</p> <p><b>Відвідання занять</b> є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Активність під час проведення лекцій і лабораторних заохочується балами при оцінюванні відповідного лабораторного завдання чи проекту. 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><b>Академічна доброчесність:</b> очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикавання джерел, списування, втручання в роботу інших студентів, здавання чужих комп’ютерних програм як своїх становлять, але</p>



	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.
<b>Опитування</b>	Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.