

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра оптоелектроніки та інформаційних технологій

Затверджено

На засіданні
кафедри оптоелектроніки та
інформаційних технологій
факультету електроніки та комп'ютерних
технологій
Львівського національного університету
імені Івана Франка
(протокол № від 2019 р.)

Завідувач кафедри:

_____ Олег КУШНІР

Силабус з навчальної дисципліни
“ Мова програмування Python ”,
що викладається в межах ОПІ
“ Інженерія програмного забезпечення ”
першого (бакалаврського) рівня вищої освіти для здобувачів з
спеціальності 121 – Інженерія програмного забезпечення

Львів 2019 р.

Назва дисципліни	Мова програмування Python
Адреса викладання дисципліни	Корпус факультету електроніки та комп'ютерних технологій, Львівський національний університет імені Івана Франка, вул. Драгоманова 50, м. Львів, 79005, вул. Ген. Тарнавського 107, м. Львів, 79011
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій Кафедра оптоелектроніки та інформаційних технологій
Галузь знань, шифр та назва спеціальності	12 – інформаційні технології 122 – Комп'ютерні науки
Викладачі дисципліни	Свелеба С.А., доктор фізико-математичних наук, старший науковий співробітник, професор кафедри оптоелектроніки та інформаційних технологій
Контактна інформація викладачів	serhiy.sveleba@lnu.edu.ua, https://electronics.lnu.edu.ua/employee/sveleba-serhij-andrijovych
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю): ауд. 213, корпус факультету електроніки та комп'ютерних технологій, вул. Ген. Тарнавського 107, м. Львів. Також можливі он-лайн консультації через MS Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
Сторінка курсу	https://moodle.elct.lnu.edu.ua/course/view.php?id=51
Коротка анотація дисципліни	Дисципліна “ Мова програмування Python ” є нормативною навчальною дисципліною з спеціальності 122 – Комп'ютерні науки для освітньої програми “Інформаційні системи та технології”, яка викладається в 4-му семестрі в обсязі 3.5 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Інформація про дисципліну	В курсі розглядаються базові поняття структурного програмування на мові Python 3.X. Формування у студентів базових понять структурного програмування на Python 3.X, вміння застосовувати їх на практиці.
Мета та цілі дисципліни	<i>Метою</i> є формування у студентів базових понять структурного програмування на Python 3.X, вміння застосовувати їх на практиці. <i>Цілі:</i> є набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень в області програмування мовою Python відповідно до освітньої кваліфікації бакалавр з інженерії програмного забезпечення.
Література для вивчення дисципліни	<ol style="list-style-type: none"> 1. Програмування числових методів мовою PYTHON / А. Ю. Дорошенко [та ін.]; за ред. А. В. Анісімова. – ВПЦ "Київський університет", 2013. – 464 с. 2. Програмування числових методів мовою Python : підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с. 3. Сучасні методи аналітичного і числового програмування в задачах теоретичної фізики: метод. рекомендації / укл.: Ю.О. Сеті. – Чернівці :

	<p>Чернівецький нац. ун-т, 2018. – 69 с.</p> <p>4. Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни “Технології розробки алгоритмів” для студентів денної та заочної форми навчання спеціальностей 6.050102 та 123 “Комп’ютерна інженерія”, 6.170103 та 125 “Кібербезпека” / уклад. Гермак В.С.; Кропивницький: ЦНТУ – 2018.– 81 с.</p> <p>5. Методичні вказівки до виконання лабораторних робіт з дисципліни “Інженерія прикладних інтелектуально-орієнтованих програмних продуктів” для студентів спеціальностей 121 “Інженерія програмного забезпечення” та 122 “Комп’ютерні науки та інформаційні технології” (всіх форм навчання) / В.М. Льовкін. – Запоріжжя : ЗНТУ, 2016. – 80 с.</p> <p>6. Downey, A.V. Think Python [Текст] / Allen V. Downey. – O’Reilly, 2012. – 300 p.</p> <p>7. Phillips, D. Python 3 Object Oriented Programming [Текст] / Dusty Phillips. – Birmingham : Packt Publishing, 2010. – 404 p.</p>
Обсяг курсу	160 години аудиторних занять. З них 32 години лекцій, 32 години лабораторних робіт та 96 година самостійної роботи
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p><i>Знати</i></p> <ul style="list-style-type: none"> • основні поняття структурного програмування; • базові вбудовані типи об’єктів структурного програмування (числа, рядки, списки, словники, кортежі, файли, множини) і синтаксичні конструкції використання цих об’єктів у вигляді літералів (виразів, які генерують ці об’єкти); • елементи структурного програмування (інструкції if, while, for); • принципи побудови функцій користувача, з використанням локальних і глобальних змінних; • організацію рекурсивних функцій користувача, освоїти методики визначення та практичного застосування рекурсивних функцій у програмах; • основні положення роботи з бінарними та текстовими файлами, модульний принцип організації програми; основи побудови сховищ даних. <p><i>Вміти</i></p> <ul style="list-style-type: none"> • використовувати базові поняття структурного програмування для розв’язання різних задач; • створювати і обробляти структури, масиви структур, найпростішу обробку файлів, створювати і використовувати функції. <p>Після вивчення даного курсу «Мова програмування Python» здобувачі набувають таких Загальних та Фахових компетентностей та Програмних результатів навчання:</p> <p>K01. Здатність до абстрактного мислення, аналізу та синтезу. K02. Здатність застосовувати знання у практичних ситуаціях. K03. Здатність спілкуватися державною мовою як усно, так і письмово. K04. Здатність спілкуватися іноземною мовою мовою як усно, так і письмово. K14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. K15. Здатність розробляти архітектури, модулі та компоненти програмних</p>

	<p>систем.</p> <p>K20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.</p> <p>K26. Здатність до алгоритмічного та логічного мислення</p> <p>ПР05: Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.</p> <p>ПР23: Вміти документувати та презентувати результати розробки програмного забезпечення.</p>
Ключові слова	Цикли, масиви, вбудовані функції, функції користувача, класи, атрибути класу, об'єкти класу
Формат курсу	Очний
Теми	<p>Тема 1. Алгоритми послідовної (лінійної) структури. Числа та операції над ними. Введення даних з клавіатури.</p> <p>Тема 2. Алгоритми розгалуженої структури (інструкція if)</p> <p>Тема 3. Алгоритми циклічної структури (інструкція while).</p> <p>Тема 4. Алгоритми циклічної структури (інструкція for) .</p> <p>Тема 5. Списки: одновимірні та двовимірні масиви. Генерація випадкових чисел.</p> <p>Тема 6. Методи. Сортування. Присвоєння .</p> <p>Тема 7. Рядки символів. Множини.</p> <p>Тема 8. Функції користувача. Наближене обчислення функцій.</p> <p>Тема 9. Рекурсивні функції. Документування коду.</p> <p>Тема 10. Словник. КORTEЖ.</p> <p>Тема 11. Файли даних. Модульний принцип організації програми.</p> <p>Тема 12. Елементи ООП.</p>
Підсумковий контроль, форма	Залік в кінці семестру
Пререквізити	<p>Для вивчення даного курсу студентам потрібні базові знання з курсів:</p> <ul style="list-style-type: none"> - дискретна математика; - теорія алгоритмів; - алгоритмізація і програмування.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Інформаційні методи (лекція, бесіда, ілюстрація, демонстрація); дедуктивні методи на основі узагальнень; евристичні методи (проблемна лекція); інтерактивні методи).
Необхідне обладнання	Комп'ютер із необхідним програмним забезпеченням, доступ до Internet мережі.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	<p>Оцінювання проводиться упродовж семестру за 100-бальною шкалою. Бали нараховуються за такими видами робіт з наступним співвідношенням:</p> <ul style="list-style-type: none"> • лабораторні роботи: 55% семестрової оцінки; максимальна кількість балів 55. • контрольні заміри (3 модулі): 45% семестрової оцінки; максимальна кількість балів 45.

	<p>Загалом упродовж семестру 100 балів.</p> <p>Академічна доброчесність: Очікується, що роботи студентів будуть оригінальними дослідженнями чи міркуваннями. Списування та втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в написанні завдань є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p> <p>Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні зайняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів робіт, передбачених курсом.</p> <p>Література. Уся література, яку 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>Політика виставлення балів. Враховуються бали, набрані при поточному контролі та бали підсумкового тестування. При цьому обов'язково враховуються присутність на заняттях та активність студента під час практичного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін.</p> <p>Жодні форми порушення академічної доброчесності не толеруються.</p>
<p>Питання до заліку чи екзамену.</p>	<p>Типи даних, які розпізнає <i>Python</i>.</p> <p>Опишіть три варіанти використання функції range. Яку функцію замість range рекомендується застосовувати у разі великого розміру діапазону? Чому це може бути важливо?</p> <p>Чи можуть списки містити інші списки як елементи?</p> <p>Яким чином може завершитися виконання циклу?</p> <p>Яким чином кодуються логічні значення в мові <i>Python</i>? Чи має <i>Python</i> окремий логічний тип?</p> <p>Які функції можна використовувати для введення й виведення даних? Оптимізації циклів.</p> <p>Які методи для виклику об'єктів-списків? Їх призначення.</p> <p>Основні характеристики функціонального програмування. Як підтримується FP мовою <i>Python</i>?</p> <p>Яким чином передаються параметри у функції <i>Python</i>?</p> <p>Для чого документувати функції?</p> <p>Якими способами можна отримати доступ до документації з мови <i>Python</i>?</p> <p>Які є можливості генерування списків у мові <i>Python</i>?</p> <p>Яким чином визначається кортеж, що містить один елемент?</p> <p>Характеристика структуру даних "словник".</p> <p>Поняття "модуль" і "пакет"? Який між ними зв'язок?</p> <p>Варіанти використання інструкції <code>import</code>?</p> <p>Чому не варто використовувати інструкцію <code>from <модуль> import *</code>?</p> <p>Яким чином здійснюється пошук модулів інтерпретатором?</p> <p>Як можна зробити модулі видимими для інтерпретатора?</p> <p>Яким чином можна отримати перелік імен, які визначено в певному модулі?</p> <p>Опишіть можливості <i>Python</i> із форматowanego виведення.</p>

	<p>Чим функція <code>str()</code> відрізняється від функції <code>repr()</code>? Функції для роботи з файлами і методи файлових об'єктів. Виняткові ситуації і яким чином здійснюється їх обробка у Python? Гілка <code>finally</code> в інструкції <code>try</code>? Чи можна її поєднувати з гілками <code>except</code>? Які два класи виняткових ситуацій наявні в Python? Який із них рекомендується використовувати у програмах? Області види мостів імен у Python. Які концепції ООП підтримує об'єктна модель Python? Які обмеження є в моделі ООП, реалізованій у Python? Механізми спадкування, реалізовані в мові Python. Оператори, що застосовуються до всіх об'єктів послідовностей. Оператори, що застосовуються тільки до списків, кортежів та словників. Вбудовані функції мови Python. Інструкції для організації циклів наявні в Python? У якому разі виконується гілка <code>else</code> для циклу <code>while</code>? Поясніть переваги використання механізму просторів імен. Як Python визначає логічне значення для об'єкта? Перерахуйте основні логічні операції. Який порядок обчислення виразів у операторі <code>and</code>? Чи можна порівнювати об'єкти довільних типів? Перерахуйте числові типи, які містить Python, і опишіть область використання кожного з них. Наведіть та опишіть арифметичні операції. Для чого можна використовувати бітові операції? Перерахуйте незмінювані послідовності, які є в мові Python. Які об'єкти можна використовувати як ключі до словника? Перерахуйте й опишіть операції над змінюваними послідовностями. Які виняткові ситуації можуть виникати при їх використанні? Опишіть викликувані об'єкти мови Python. Поясніть переваги використання спеціальних методів класів у мові Python. Дайте огляд бібліотечних модулів мови Python. Для чого може використовуватись переведення об'єктів у послідовну форму (серіалізація)?</p>
Опитування	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.</p>

**Схема курсу “ Мова програмування Python ”
для студентів спеціальності 122 – Комп’ютерні науки**

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)* *лекція, самостійна, дискусія, групова робота)	Література.** * Ресурси в інтернеті	Завдання, год	Термін виконання
1	Алгоритми послідовної (лінійної) структури. Числа та операції над ними. Введення даних з клавіатури. Алгоритми послідовної (лінійної) структури, з процедурами запуску програм, які реалізують ці алгоритми на мові Python. Python: запуск програми користувачем. Числа та операції над ними. Введення даних з клавіатури	Лекція	[1], [2], [3], [6], [7]	2	кінець поточного тижня
1	Лаб.1 Розробка блок-схеми алгоритму розв’язання задачі. Обчислити вирази.	Лабораторна робота		2	кінець поточного тижня
2	Алгоритми розгалуженої структури (інструкція if) . Алгоритми розгалуженої структури та їх реалізація. Динамічна типизація. Умовна інструкція if. Реалізація обробки помилок (виключень у вигляді рядків): за допомогою інструкції try; перевірка змісту рядку введення за допомогою методу isdigit.	Лекція	[1], [3], [6],	2	кінець поточного тижня
2	Лаб.2. а)Алгоритми розгалуженої структури (інструкція if). б)Алгоритми циклічної структури (інструкція while). в) Алгоритми циклічної структури (інструкція for).	Лабораторна робота		2	кінець поточного тижня
3	Алгоритми циклічної структури (інструкція while). Алгоритми циклічної структури, їх реалізацією на мові Python. Оператор циклу while. Тип даних «рядки».	Лекція	[1], [2], [5]	2	кінець поточного тижня
3	Лаб.3. а) Функції користувача. Наближене обчислення функцій. б) Рекурсивні функції. Документування коду.	Лабораторна робота		2	кінець поточного тижня
4	Алгоритми циклічної структури (інструкція for). Алгоритми циклічної структури на основі оператора циклу for, Модуль itertools (функція itertools.count), функція range, ітераційні процеси.	Лекція	[1], [2], [7]	2	кінець поточного тижня
4	Лаб.4 Рядки символів. Завдання 1 - обробка рядків. Завдання 2 – обробка речення.	Лабораторна робота		2	кінець поточного тижня
5	Списки: одновимірні та двовимірні масиви. Генерація випадкових чисел. Одновимірні та двовимірні масиви, структура організації масивів та способи доступу до їх елементів.	Лекція	[1], [2], [3]	2	кінець поточного тижня

	Списки, масиви даних, процедура генерації випадкових чисел (модуль random)				
5	Лаб.5. Списки: одновимірні та двовимірні масиви. Генерація випадкових чисел. Завдання 1. Одновимірні масиви (вектори). Завдання 2. Двовимірні масиви (матриці).	Лабораторна робота		2	кінець поточного тижня
6	Методи. Сортвання. Присвоєння. Методи додавання елемента « ? » у список; додавання списку у список. Додавання списку елементів у вказану позицію. Визначення зсуву елемента за заданим значенням. Підрахунок кількості елементів. Сортвання. Зміна порядку слідування елементів. Зменшення списку. Видалення елемента. Видалення групи елементів. Присвоєння за індексом. Присвоювання зрізу значень. Генератори списків.	Лекція	[1], [2], [3]	2	кінець поточного тижня
6	Модуль 1.	Модуль		2	кінець поточного тижня
7	Рядки символів. Множини. Множина, рядки символів. Обробка рядків. Використання множин. Пошук перетину двох послідовностей у вигляді рядків. Використання множини для визначення перетину послідовностей у вигляді рядків. Створення множин на основі генераторів множин. Використання конструктору frozenset. Операції над множинами, записані у вигляді бінарних операцій.	Лекція	[1], [2], [3], [5],	2	кінець поточного тижня
7	Лаб.6. Рядки символів. Множини.	Лабораторна робота		2	кінець поточного тижня
8	Функції користувача. Наближене обчислення функцій. Основні складові функцій, оголошення та опис функцій. Оператор форматування %. Інструкції, які використовують при створенні функцій.	Лекція	[1], [2], [6], [7],	2	кінець поточного тижня
8	Лаб. 7. Словник. Кортеж	Лабораторна робота		2	кінець поточного тижня
9	Рекурсивні функції. Документування коду. Організація рекурсивних функцій користувача, методики визначення та практичного застосування рекурсивних функцій у програмах, ефективний засіб реалізації циклічних алгоритмів. Документування коду. Вбудовані рядки документування. Lambda-вирази.	Лекція	[1], [2], [3], [7]	4	Кінець поточного тижня
9	Лаб.8 Файли даних. Модульний принцип організації програми. Завдання 1: бінарні файли. Завдання 2:	Лабораторна робота		2	кінець поточного тижня

	текстові файли.				
10	Словник. Кортеж. Словники, кортежі мови Python. Створення словників. Операції зі словниками. Використання collections.Counter. Використання lambda-виразів. Літерали словників та операції. Методи словників, літерали кортежів та операції.	Лекція	[1], [2], [3],	4	кінець поточного тижня
10	Модуль 2	Модуль		2	кінець поточного тижня
11	Файли даних. Модульний принцип організації програми. Основні положення роботи з файлами на мові Python. Бінарні та текстові файли, модульний принцип організації програми. Файл. Модуль. Відкриття файлу для зчитування. Використання функції os.path.join для побудови шляху до файлу. Запис даних у текстовий файл. Використання оператора with для закриття файлу. Відкриття текстового файлу для зчитування з вказівкою кодування <code>_file_</code> . Відкриття файлу для зчитування та запису. Відкриття файлу для до запису. Перезапис файлу. Використання файлового об'єкта io.StringIO. Використання бінарного файлу. Використання json. Серіалізація – десеріалізація. Серіалізація за допомогою pickle. Декоратор.	Лекція	[1], [2], [7],	4	кінець поточного тижня
11	Лаб.9. Класи та об'єкти – екземпляри класу.	Лабораторна робота		2	кінець поточного тижня
12	Елементи ООП. Основні поняття. Абстракція і деконпозиція. Об'єкти. Типи і класи. Визначення класу. Інкапсуляція. Доступ до властивостей. Приховування даних. Поліморфізм. Імітація типів. Відносини між класами. Спадкування. Множинне спадкування. Порядок дозволу методів. Агрегація. Контейнери. Ітератори. Асоціація. Статичний метод. Метод класу. Метакласи. Мультиметод. Стійкі об'єкти.	Лекція	[1], [2], [4], [6],	4	кінець поточного тижня
12	Лаб.10. Побудова Фреймової моделі. Наслідування атрибутів класу в ООП.	Лабораторна робота		2	кінець поточного тижня
13	Лаб.11 ООП: розробка сховища даних	Лабораторна робота		2	кінець поточного тижня
15	Модуль 3	Модуль		2	кінець поточного тижня