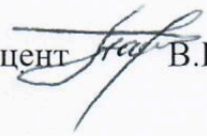


Міністерство освіти і науки України  
Львівський національний університет природокористування  
Факультет механіки, енергетики та інформаційних технологій  
Кафедра інформаційних технологій



**ЗАТВЕРДЖЕНО**

Гарант освітньо-професійної програми «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти

к.т.н., доцент  В.В. Пташник

**СИЛАБУС**  
**НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**  
**«Навчальна практика (2 курс)»**

освітньо-професійна програма «Комп'ютерні науки»  
спеціальність 122 «Комп'ютерні науки»  
перший (бакалаврський) рівень вищої освіти

**ВИКЛАДАЧ**



**Луб Павло Миронович**

Електронна пошта:

*pollylub@ukr.net*

Телефон

+380961606701

Доцент кафедри інформаційних технологій Львівського національного університету природокористування, кандидат технічних наук. Викладач з 22-річним досвідом, автор та співавтор понад 150 наукових праць, чотирьох посібників, трьох монографій, понад 50 навчально-методичних розробок.

Читає курси: Алгоритмізація та програмування, Автоматизовані системи підтримки прийняття рішень, Управління ІТ-проектами, Моделювання систем, Імітаційне моделювання, ІТ інструменти SMM та SERM, Основи інженерного менеджменту.

Сфера наукових інтересів: моделювання адаптивних технологічних систем рільництва, проектно-технологічні основи інженерії систем збирання врожаю.

**ЛЬВІВ 2024**

**Галузь знань: 12 Інформаційні технології**  
**Спеціальність: 122 «Комп'ютерні науки»**  
**Освітньо-професійна програма «Комп'ютерні науки»**  
**Рівень вищої освіти – перший (бакалаврський)**  
**Кількість кредитів – 6**  
**Рік підготовки, семестр – 2 рік, 4 семестр**  
**Компонент освітньої програми: обов'язкова**  
**Мова викладання: українська**

## ОПИС ДИСЦИПЛІНИ

Навчальна компонента розроблено таким чином, щоб надати студентам необхідні навички, обов'язкові для того, щоб поглибити та закріпити отримані теоретичні знання з основних курсів. Тому, під час Навчальної практики студенти вивчають мови програмування, вчать оцінювати складність алгоритмів і програм, тестувати їх, застосовувати системні та програмовані методи налагодження програм, знайомляться із базовими підходами до розробки програмного забезпечення, що необхідно для самостійного виконання індивідуальних завдань, які полягають в проектуванні, розробці програмних засобів та відповідної супровідної документації.

**Міждисциплінарні зв'язки:** Для вивчення курсу студенти потребують базових знань з дисциплін «Вища математика», «Дискретна математика», «Чисельні методи», «Іноземна мова», «Алгоритми та структури даних», «Програмування», «Об'єктно-орієнтоване програмування».

Вимоги до знань та умінь визначаються галузевими стандартами вищої освіти України.

**Предметом вивчення освітньої компоненти «Навчальна практика»** є процес навчання і підготовки фахівця за освітньо-професійною програмою «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти, який дозволить застосовувати знання із базових алгоритмічних структур і базових структур даних з використанням сучасних технологій розроблення програмного забезпечення.

**Мета навчальної дисципліни.** Метою вивчення нормативної компоненти «Навчальна практика» є отримання студентами практичних навичок роботи з сучасними інформаційно-технічними засобами, знайомство з технологіями розробки, упровадження та експлуатації програмного та інформаційного забезпечення, закріплення та поглиблення теоретичних та практичних знань отриманих студентами при вивченні спеціальних дисциплін, набуття навичок для самостійного написання програм відповідної структури.

**Основними завданнями освітньої компоненти «Навчальна практика»** є: основні поняття програмування, принципи розробки програми, типові алгоритмічні конструкції, основні структури даних; основи структурного та функційного програмування; основи об'єктно-орієнтованого програмування. В результаті вивчення дисципліни студенти розвивають уміння аналізувати поставлену задачу, обирати доцільний метод її розв'язання; складати алгоритм вирішення задачі; писати та підлагоджувати програми, володіти сучасними засобами розроблення програмного забезпечення.

## СТРУКТУРА КУРСУ

1. Організаційний етап.
  - 1.1. Проведення загальних організаційних зборів практикантів та первинного інструктажу з техніки безпеки.
  - 1.2. Узгодження, видача завдань та календарних термінів проходження практики.
2. Практичний етап.
  - 2.1. Ознайомлення із обладнанням, документацією, режимами виконання завдання.
  - 2.2. Виконання індивідуального завдання.
3. Завершальний етап.
  - 3.1. Обговорення результатів проведення Навчальної практики.
  - 3.2. Підготовка звіту і щоденника за результатами проходження Навчальної практики.
  - 3.3. Представлення та корегування розробленого звіту.
  - 3.4. Захист звіту із проходження практики.

Зміст індивідуальних завдань конкретизується під час проходження Навчальної практики, зібрані матеріали мають бути використані для підготовки рефератів, доповідей, курсових та дипломних проектів.

Індивідуальні завдання повинні бути орієнтовані на аналіз роботи та вивчення особливостей мов програмування, фреймворків, інформаційних систем тощо.

Індивідуальні завдання видаються керівником практики перед її початком та оформлюються окремою сторінкою у звіті з проходження практики за підписом керівника.

**Тривалість практики – 4 тижні.**

## НАВЧАЛЬНИЙ КОНТЕНТ

Тема	Результати навчання
<b>Інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією – Python</b>	
Тема 1. Прості типи даних	Об'єкти та змінні; Числа; Цілі числа; Перетворення типів: функція int(); Пріоритет операторів; Числа з плаваючою крапкою; Перетворення типів: функція float(); Математичні функції
Тема 2. Рядки	Створення рядків і функція print(); Стиль форматування: функція format(), f-рядки; Перетворення типів: функція str(); Керуючі символи; Конкатенація рядків; Дублювання рядків; Доступ до елемента рядка за індексом; Зрізи: функція slice[start: end: step]; Довжина рядка: функція len(); Розділення рядка: функція split(); Об'єднання рядків: функція join(); Регістр і вирівнювання; Заміна символів: функція replace(); Введення даних з клавіатури; Функція input()
Тема 3. Списки і кортежі	Створення списків; Довжина списку; Перетворення типів: функція list(); Доступ до елементів списку; Списки списків; Зміна елементів списку; Розділення і зворотний порядок елементів списку; Додавання елемента у список
Тема 4. Об'єднання списків	Видалення елементів зі списку; Чи є елемент у списку?; Кількість значень у списку; Сортування списків; Список у зворотному порядку; Створення числових списків; Кортежі

Тема 5. Словники і множини	Створення словника; Перетворення типів: функція dict(); Додавання і зміна елементів словника; Об'єднання словників; Видалення елементів із словника; Чи є ключ у словнику?; Значення із словника; Ключі і значення словника; Обробка відсутніх ключів словника: функція.setdefault(); Множини; Складені структури даних
Тема 6. Структури коду	Створення і перевірка умов; Булеві значення; Оператори порівнювання; Булеві оператори; Поєднання булевих значень, операторів порівнювання, булевих операторів; Вказівка розгалуження; Що істина, а що хибність?; Команда if; Команда else; Команда elif; Конструкція if/elif/else
Тема 7. Вказівка повторення	Команда while; Переривання циклу, break; Нескінченний цикл і вихід з нього; Продовження циклу, continue; Цикл for; Функція range(); Цикл for і послідовності; Функція enumerate(); Функція zip(); Функція map(); Включення (скорочення синтаксису); Включення для списків; Включення для словників; Генератори
Тема 8. Функції	Визначення і виклик функцій; Значення None; Позиційні та іменовані аргументи; Значення за замовчуванням; Використання аргументів з символами * і **; Анонімні функції: інструкція lambda; Обробка помилок; Виняткові ситуації; Блок try-except
Тема 9. Модулі і пакети	Простір імен і області видимості; Імпорт модулів: інструкція import; Модуль random: правила імпорту; Модуль math: що всередині?; Стандартна бібліотека Python: короткий огляд модулів; Модулі string і inspect, функції capwords(), isinstance(), getmembers() і repr(); Модуль collections, функція OrderedDict(); Модуль pprint, функція pprint(); Модуль decimal, функції Decimal(), getcontext(); Модуль __name__; Аргументи командного рядка; Пакети; Створення власних пакетів; Менеджер пакетів pip; Віртуальні середовища
Тема 10. Файли	Відкриття текстового файлу; Запис даних у текстові файли; Зчитування даних з текстових файлів; Інструкція with; Бінарні файли; Структуровані текстові файли; CSV; XML; JSON; Файли бази даних; PythonDatabase API; Мова запитів SQL; SQLite
Тема 11. Класи та об'єкти	Створення і використання класу; Створення класу; Створення екземпляру класу; Доступ до атрибутів; Виклик методів; Створення декількох екземплярів; Робота з класами та екземплярами; Присвоювання атрибуту значення за замовчуванням; Зміна значень атрибутів; Наслідування; Перевизначення методу; Екземпляри як атрибути
<b>Динамічна, об'єктно-орієнтована прототипна мова програмування JavaScript</b>	
Тема 1. Основи JavaScript	Структура коду; Сучасний режим, "usestrict"; Змінні; Типи даних; Взаємодія: alert, prompt, confirm; Перетворення типу; Базові оператори, математика; Оператори порівняння; Умовні розгалуження: if, '?'; Логічні оператори; Оператор об'єднання з null '??'; Цикли: while і for; Конструкція "switch"; Функції; Функціональні вирази; Стрілкові функції, основи
Тема 2. Якість коду	Налагодження в браузері; Стандарт оформлення коду; Коментарі; Ніндзя-код; Автоматичне тестування за допомогою Mocha; Поліфіли та транспілятори
Тема 3. Об'єкти: основи	Об'єкти; Копіювання об'єктів та посилання; Збирання сміття; Методи об'єкта, "this"; Конструктори, оператор "new"; Опціональний ланцюжок '?.'; Тип даних Символ; Перетворення об'єктів в примітиви
Тема 4. Типи даних	Методи примітивів; Числа; Рядки; Масиви; Методи масивів; Ітеративні об'єкти; Map та Set; WeakMap та WeakSet; Object.keys, values, entries; Деструктуроване присвоєння; Дата і час; Методи JSON, toJSON

Тема 5. Розширена робота з функціями	Рекурсія та стек; Залишкові параметри та синтаксис поширення; Область видимості змінної, замикання; Застаріле ключове слово "var"; Глобальний об'єкт; Об'єкт функції, NFE; Синтаксис "newFunction"; Планування: setTimeout та setInterval; Декоратори та переадресація виклику, call/apply; Прив'язка контексту до функції; Повторення стрілкових функцій
Тема 6. Властивості об'єкта, їх конфігурація	Прапори та дескриптори властивостей; Гетери і сетери властивостей
Тема 7. Прототипи, наслідування	Успадкування через прототипи; F.prototype; Вбудовані прототипи; Методи прототипів, об'єкти без proto
Тема 8. Класи	Базовий синтаксис класу; Наслідування класу; Статичні властивості та методи; Приватні та захищені властивості та методи; Розширення вбудованих класів; Перевірка класу: "instanceof"; Міксини
Тема 9. Обробка помилок	Робота з помилками, "try...catch"; Нестандартні помилки, розширення Error
Тема 10. Проміси, async/await	Введення: колбеки; Проміси; Ланцюжок промісів; Проміси: обробка помилок; Promise API; Промісифікація; Мікрозадачі; Async/await
Тема 11. Генератори, розширена ітерація	Генератори; Асинхронні ітератори та генератори
Тема 12. Модулі	Вступ до модулів; Експорт та імпорт; Динамічні імпорти

### ФОРМУВАННЯ ПРОГРАМНИХ КОМПЕТЕНТНОСТЕЙ

Індекс в матриці ОПШ	Програмні компоненти
ЗК2	Здатність застосовувати знання у практичних ситуаціях.
ЗК3	Знання та розуміння предметної області та розуміння професійної діяльності
ЗК7	Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
СК8	Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління
ПРН5	Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.
ПРН9	Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.
ПРН14	Знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

## РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ

Поточна перевірка та самостійна робота (разом 100 балів)											Сума (100 балів)	
Інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією – <i>Python</i>												
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11		50
4	4	4	5	5	5	4	4	5	5	5		
Динамічна, об'єктно-орієнтована прототипна мова програмування <b>JavaScript</b>											50	
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11		T12
4	4	4	4	4	5	4	4	4	4	4	5	

T1, T2 ... T12 – теми

## ЛІТЕРАТУРНІ ДЖЕРЕЛА

1. Олексій Васильєв. Програмування мовою Python. Київ: Навчальна книга. Богдан, 2019. 504 с.
2. Lambert K. A. Fundamentals of Python: First Programs, 2nd Edition. Cengage, 2019. 498 p.
3. Підручник мови Python. URL: [https://uk.wikibooks.org/wiki/Підручник\\_мови\\_Python](https://uk.wikibooks.org/wiki/Підручник_мови_Python) 9. <https://docs.python.org/3/library/index.html>
4. Путівник мовою програмування Python. URL: [https://pythonguide.rozh2sch.org.ua/#\\_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%96\\_%D1%82%D0%B8%D0%BF%D0%B8\\_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85](https://pythonguide.rozh2sch.org.ua/#_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%96_%D1%82%D0%B8%D0%BF%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)
5. Чураков, А.Я. Архітектура ЕОМ [Текст]: посібник/ А.Я. Чураков, С.В. Шаров, О.В. Строкань. Мелітополь: РВЦ МДПУ, 2012. 195 с.
6. Мова програмування JavaScript. URL: <https://uk.javascript.info/getting-started>

## ІНФОРМАЦІЙНІ РЕСУРСИ

1. Бібліотечно-інформаційні ресурси – книжковий фонд, періодика та фонди на електронних носіях бібліотеки ЛНУП, державних органів науково-технічної інформації, наукових, науково-технічних бібліотек та інших наукових бібліотек України.
2. Електронний ресурс стосовно ІТ компаній Львівського регіону: [http://www.invest-lvivregion.com/it-компанії\\_ua\\_285cms.htm](http://www.invest-lvivregion.com/it-компанії_ua_285cms.htm)
3. Портал об'єктно-орієнтованого програмування: <http://oop.in.ua/tag/FAQ/>
4. Книжки з програмування: як читати і що саме: <https://dou.ua/lenta/articles/programming-books/>
5. Українська технічна література. Програмування: <https://ukrtechlibrary.wordpress.com/tag/програмування/>
6. Задачі програмування із прикладами розв'язку: <http://purecodecpp.com/uk/archives/433>
7. Задачі програмування із прикладами розв'язку: <http://library.nuft.edu.ua/ebook/datathree.php?ID=138>
8. Задачі програмування із прикладами розв'язку: <http://abramov.org.ua/blog/category/opp/obchislennya-%D1%96z-zber%D1%96gannyam-posl%D1%96dovnosti/>

### **ПОЛІТИКА ОЦІНЮВАННЯ**

**Політика щодо дедлайнів та перескладання:** Звіти, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (75% від можливої максимальної кількості балів за вид діяльності балів). Перескладання звіту із Навчальної практики відбувається за поважних причин (наприклад, лікарняний).

**Політика щодо академічної доброчесності:** Списування під час виконання завдань заборонені (в т.ч. із використанням мобільних девайсів). Мобільні пристрої дозволяється використовувати лише під час он-лайн тестування та підготовки практичних завдань під час практики.

**Політика щодо відвідування:** Відвідування занять є обов'язковим компонентом оцінювання. За об'єктивних причин (наприклад, хвороба, працевлаштування, міжнародне стажування) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу.

**До Силабусу також готуються матеріали навчально-методичного комплексу:**

- 1) Навчальний контент;
- 2) Тематика та зміст практичних завдань;
- 3) Завдання для індивідуальної роботи тощо;
- 4) Електронне навчання у віртуальному навчальному середовищі ЛНУП (<https://moodle.lnup.edu.ua/>).