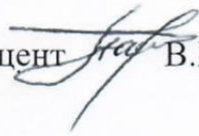


Міністерство освіти і науки України
Львівський національний університет природокористування
Факультет механіки, енергетики та інформаційних технологій
Кафедра інформаційних технологій



ЗАТВЕРДЖЕНО

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

к.т.н., доцент  В.В. Пташник

**СИЛАБУС
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ПРОГРАМУВАННЯ»**

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 систем та технологій Львівського національного університету природокористування, кандидат технічних наук. Викладач з 21-річним досвідом, автор та співавтор понад 140 наукових праць, чотирьох посібників, трьох монографій, понад 45 навчально-методичних розробок.

Читає курси: Алгоритмізація та програмування, Автоматизовані системи підтримки прийняття рішень, Управління проектами, Інформаційний маркетинг та менеджмент. Сфера наукових інтересів: моделювання адаптивних технологічних систем рільництва, проектно-технологічні основи інженерії систем збирання технічних культур.

ЛЬВІВ 2023

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Рівень вищої освіти – перший (бакалаврський)

Кількість кредитів – 9

Рік підготовки, семестр – 1 рік, 2 семестр, 2 рік 3 семестр

Компонент освітньої програми: обов'язкова

Мова викладання: українська

Опис дисципліни

Під час розвитку процесорів для обчислювальних машин було досягнуто фізичних границь збільшення їх частоти. Тому в останнє десятиліття стрімко зростає кількість та швидкими темпами розвиваються технології паралельних та розподілених обчислень.

Програма дисципліни «Технології паралельних та розподілених обчислень» відноситься до дисциплін професійної підготовки та складена відповідно до освітньо-професійної програми «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти.

Міждисциплінарні зв'язки: освітня компонента «Програмування» є складовою частиною циклу професійної підготовки для здобувачів освітньо-професійної програми «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти. Вивчення дисципліни передбачає наявність систематичних та ґрунтовних знань із суміжних курсів – «Алгоритми та структури даних».

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

Предметом вивчення освітньої компоненти «Програмування» є загальні поняття про мови програмування, парадигми програмування: імперативне, функційне та об'єктно-орієнтоване. Це все представляється на прикладі мови Пайтон.

Метою вивчення освітньої компоненти Мета – формування компетенцій, необхідних для ефективного використання бібліотек мови Python для вирішення прикладних професійних 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

Основними завданнями освітньої компоненти «Програмування» є: набуття теоретичних знань і практичних навичок в області створення алгоритмів та програм мовою Пайтон, оволодіння концепціями сучасного програмування на Пайтоні з використанням бібліотек Numpy, Pandas, Matplotlib та інших.

Навчальний контент

Структура курсу

Тижд.	Тема, план, короткі тези	Форма заняття	Тривалість, год	Термін виконання
1.1	Що таке "комп'ютер"? Кодування інформації. Мови програмування. Структура програми мовою C++, include, main(). Стандартні заголовкові файли, порядок трансляції.	Лекція	2	
	Системи числення: позиційні, непозиційні. Основа системи числення. Двійкова система: алгоритми переведення, арифметика. Кодування від'ємних чисел.	Лабораторна робота	2	Наступне лабораторне заняття
1.2	Система типів. Логічний тип: зображення, оператори, вирази. Літерний тип. Зображення літер. Цілі типи: назви, граничні значення, зображення, оператори. Оператори присвоєння. Дійсні типи. Перетворення типів. Математичні функції. Задача про трикутник. Використання cin, cout.	Лекція	2	
	Вісімкова, шістнадцяткова системи числення: переведення, арифметика, зв'язок з двійковою. Контрольна робота "Системи числення".	Контрольна робота	2	
1.3	Послідовні та галужені алгоритми. Інструкції if, if else, switch, break. Задача max(a,b,c). Задача "Ми знайшли k грибів". Оператор ?:. Повторювані алгоритми.	Лекція	2	

	Інструкції циклу for, while, do while. Оголошення в інструкціях. Задача про кількість цифр числа. Введення послідовностей чисел (стан потоку введення). Інструкції break, continue, goto.			
	Програмна реалізація простих послідовних алгоритмів: обчислення за математичними формулами, найпростіші задачі цілочислової арифметики. Правила зображення цілих і дійсних чисел. Запис арифметичних і логічних виразів, використання стандартних функцій. Оформлення введення, виведення даних.	Лабораторна робота	2	Наступне лабораторне заняття
1.4	Модифікатор const, тип перелік. Загальна структура оголошення, typedef. Структуровані типи даних: масиви, рядки. Введення-виведення рядків. Функції для роботи з рядками (cstring). Використання масивів для зберігання/побудови послідовності значень.	Лекція	2	
	Програмна реалізація галужених алгоритмів. Використання умовних інструкцій: повної, вкороченої, вкладеної. Використання інструкції вибору варіанту.	Лабораторна робота	2	Наступне лабораторне заняття
1.5	Вказівники, арифметика вказівників, застосування до масивів. Динамічні змінні: створення, використання, знищення. Введення-виведення масивів. Робота з двохвимірними масивами. Побудова лінійного списку за допомогою вказівників.	Лекція	2	
	Програмна реалізація повторюваних алгоритмів. Прості цикли. Взаємозамінність різних інструкцій циклу. Опрацювання послідовностей значень без зберігання у пам'яті та з використанням статичних масивів.	Лабораторна робота	2	Наступне лабораторне заняття
1.6	Модульне програмування. Функції в C++: прототип, визначення, виклик. Параметри-значення, параметри-вказівники, вказівники і специфікатор const. Тип посилання, використання в параметрах функцій.	Лекція	2	
	Поєднання цикла з галуженням. Вкладені цикли (у матричних задачах). Контрольна робота "Основи алгоритмізації"	Контрольна робота	2	
1.7	Функції і масиви. Впорядкування елементів масиву. Вказівник на функцію, масиви функцій, функції вищих порядків. Використання лямбда-виразів.	Лекція	2	
	Створення, використання, знищення динамічних одно- та двохвимірних масивів (у тому числі масивів літер). Оголошення функцій для обчислень виразів, для роботи з масивами.	Лабораторна робота	2	Наступне лабораторне заняття
1.8	Структуровані типи даних C++: структури, бітові поля, об'єднання. Конструктори, деструктори структур, селектор імені. Перевантаження операторів: арифметичних, введення-виведення. Функції для опрацювання структур.	Лекція	2	
	Оголошення і використання функцій, що змінюють свої аргументи: параметри-вказівники та параметри-посилання. Використання вказівника на функцію: алгоритми числового інтегрування, наближене розв'язування рівнянь.	Лабораторна робота	2	Наступне лабораторне заняття
1.9	Поняття зв'язної структури даних. Тип для моделювання ланок лінійного однозв'язного списку. Побудова списку з послідовності значень. Перебір елементів списку. Вставляння значення у впорядкований список. Вилучення списку з динамічної пам'яті. Сортування списком. Алгоритм злиття впорядкованих послідовностей.	Лекція	2	
	Оголошення і використання типу struct: конструктори, оператори введення-виведення. Моделювання одно- та двохзв'язного списку. Функція побудови списку, виведення, пошуку значення тощо.	Лабораторна робота	2	Наступне лабораторне заняття
1.10	Структура даних дерево. Поняття рекурсії. Рекурсивні розв'язки, рекурсивні функції. Алгоритми обходу двійкового дерева. Дерева пошуку. Сортування деревом. Збалансовані дерева, вставка, вилучення.	Лекція	2	
	Моделювання двійкового дерева, дерева пошуку. Рекурсивні та нерекурсивні алгоритми обходу дерева.	Лабораторна робота	2	Наступне лабораторне заняття
1.11	Введення/виведення даних і використання файлів у C++. Потоки, буфери, файли. Робота з багатьма файлами. Текстові та двійкові файли.	Лекція	2	
	Впорядкування масиву. Впорядкування списку. Застосування різних алгоритмів. Перевірка правильності виконання.	Лабораторна робота	2	Наступне лабораторне заняття
1.12	Аргументи функцій за замовчуванням. Статичні змінні функції. Поліморфізм і перевантаження	Лекція	2	

	функцій. Вбудовані функції. Оголошення та використання просторів імен. Класи пам'яті.			
	Завантаження колекції значень з файла. Зберігання даних до файла. Функції з параметрами за замовчуванням.	Лабораторна робота	2	Наступне лабораторне заняття
1.13	Клас - тип, оголошений користувачем. Термінологія. Прихована і доступна частини класу. Конструктори: за замовчуванням, з параметрами. Клас "Пакет акцій". Розташування програмного коду у файлах.	Лекція	2	
	Оголошення простору імен, розташування в ньому типів, функцій. Використання. Контрольна робота "Модульне програмування"	Контрольна робота	2	
1.14	Оголошення класу і визначення методів класу. Використання дружніх функцій. Перевантаження операторів: різні способи. Перетворення типів: конструктори, оператори. Клас "Time"	Лекція	2	
	Оголошення та використання класу. Конструктори з параметрами та за замовчуванням. Перевантаження операторів.	Лабораторна робота	2	Наступне лабораторне заняття
1.15	Обробка помилок за допомогою винятків: захищений блок, перехоплення, запуск, повторний запуск винятків. Розгортання стека. Стандартні класи винятків C++. Типи винятків, оголошені користувачем	Лекція	2	
	Об'єкти, що використовують рядки. Визначення конструктора копіювання, перевизначення оператора присвоєння.	Лабораторна робота	2	Наступне лабораторне заняття
1.16	Модульне тестування. Як додати до рішення проект модульного тестування. Структура класу тестів. Методи-тести. Різновиди статичних методів класу Assert.	Лекція	2	
	Контрольна робота "Об'єктне програмування"	Контрольна робота	2	
	Всього	Лекцій Лаб занять	32 32	
2.1	Ще раз про оголошення класу. Приватні методи. Клас "прямокутник". Наслідування типів, видимість членів типу. Відношення "is-a". Клас "квадрат". Конструктори підкласів. Розширені правила сумісності. Поліморфізм вказівників, посилань. Перевизначення методів, поліморфізм поведінки. Віртуальні методи та пізні зв'язування.	Лекція	2	
	Оголошення класу, що містить залежні поля та приватні методи. Визначення всіх конструкторів та деструктора. Використання статичних полів і статичних методів. Модульне тестування написаного коду.	Лабораторна робота	2	Наступне лабораторне заняття
2.2	Деструктори підкласів. Оператори введення-виведення для ієрархії класів. Класифікація об'єктів. Ієрархія класів плоских геометричних фігур. Абстрактний базовий клас, абстрактні методи. Класи "квадрат", "круг", "трикутник".	Лекція	2	
	Оголошення простої ієрархії "надклас-підклас" з віртуальними методами. Побудова та опрацювання поліморфної колекції (масиву) об'єктів. Уведення об'єктів з консолі, з файла, виведення на консоль і до файла. Надсилання поліморфних повідомлень.	Лабораторна робота	2	Наступне лабораторне заняття
2.3	Ієрархія об'ємних геометричних фігур. Відношення "has-a", композиція об'єктів. Використання вкладених класів, підкласи для розширення можливостей надкласу.	Лекція	2	
	Оголошення розгалуженої ієрархії класів. Використання абстрактного базового класу. Віртуальні методи, віртуальні деструктори.	Лабораторна робота	2	Наступне лабораторне заняття
2.4	Агрегація об'єктів. Проектування динамічного масиву об'ємних фігур. Оператори індексування. Питання власності агрегованих об'єктів. Глибоке копіювання контейнера. Бібліотека RTTI: перевірка типу об'єкта, приведення типу.	Лекція	2	
	Побудова типів на основі композиції класів (стандартних і власних). Використання власних (вкладених) класів винятків.	Лабораторна робота	2	Наступне лабораторне заняття
2.5	Закрите наслідування - ще один спосіб моделювання "has-a". Синтаксис. Порівняння можливостей з композицією. Захищене наслідування.	Лекція	2	
	Побудова типів за допомогою закритого та захищеного наслідування. Використання власних (вкладених) класів винятків.	Лабораторна робота	2	Наступне лабораторне заняття

2.6	Множинне наслідування. Випадки безпроблемного використання. Випадок спільного наднадкласу: віртуальні базові класи, конструктори, пам'ять об'єкта.	Лекція	2	
	Створення та використання класів-контейнерів. Використання засобів RTTI. Модульне тестування написаного коду.	Лабораторна робота	2	Наступне лабораторне заняття
2.7	Узагальнення в програмуванні (змінна, функція, шаблон). Шаблон функції, параметризований типом даних: оголошення, використання, явна спеціалізація, перевантаження. Шаблон класу з параметром типом (стек): оголошення, використання, явна спеціалізація. Шаблон з параметром не типом.	Лекція	2	
	Створення класів за допомогою множинного наслідування. Модульне тестування написаного коду.	Лабораторна робота	2	Наступне лабораторне заняття
2.8	Алгоритм пошуку значення в послідовності: узагальнення типу даних, структури послідовності, умови відшукання. Наслідування, включення, часткова спеціалізація. Шаблиони - параметри шаблонів. Дружні конструкції.	Лекція	2	
	Побудова та використання шаблонів функцій, наприклад, для опрацювання числових масивів. Перевантаження шаблонів. Явна спеціалізація шаблонів для окремих типів даних.	Лабораторна робота	2	Наступне лабораторне заняття
2.9	Стандартна бібліотека шаблонів (STL). Базові поняття: контейнер, ітератор, алгоритм, функтор. Класифікація ітераторів, моделі ітераторів. Класифікація контейнерів, спільні можливості.	Лекція	2	
	Побудова шаблонів класів, наприклад, для реалізації динамічних масивів. Використання зі стандартними та власними типами даних. Використання числових параметрів шаблону.	Лабораторна робота	2	Наступне лабораторне заняття
2.10	Послідовні контейнери: концепція, влаштування, базові операції. Приклади використання контейнерів vector, list, deque. Порівняння ефективності послідовних контейнерів. Ітератори вставляння. Поточкові ітератори.	Лекція	2	
	Побудова шаблонів класів, наприклад, для реалізації зв'язних списків. Створення і використання шаблону стека, параметром якого є шаблон послідовного контейнера (динамічного масиву чи зв'язного списку).	Лабораторна робота	2	Наступне лабораторне заняття
2.11	Контейнерні адаптери stack, queue, priority_queue. Приклади використання: побудова польського запису, моделювання багатопотокового обслуговування черги клієнтів. Доступ до реалізації. Наслідування від стандартних контейнерів.	Лекція	2	
	Вивчення функціональності vector<T>: дописування, вставляння, вилучення, індексування, перебір, копіювання; дослідження типу. Використання у власних програмах.	Лабораторна робота	2	Наступне лабораторне заняття
2.12	Використання класів характеристик для побудови шаблонів Класи, що налаштовують функціональність шаблону. Зміна характеристик класу string. Мета-програмування.	Лекція	2	
	Вивчення функціональності deque<T>, list<T>. Використання у власних програмах. Побудова класів, що включають послідовні контейнери. Побудова класів, що наслідують послідовні контейнери.	Лабораторна робота	2	Наступне лабораторне заняття
2.13	Класифікація функторів. Об'єкт-функція. Стандартні функтори та функціональні адаптери. Створення власних функторів. Огляд бібліотеки алгоритмів. Приклади використання для написання "програм без циклів".	Лекція	2	
	Оголошення об'єкт-функції. Використання стандартних і власних функторів і стандартних алгоритмів для опрацювання послідовних контейнерів, файлів.	Лабораторна робота	2	Наступне лабораторне заняття
2.14	Асоціативні контейнери. Спільні можливості, особливості реалізації. Різні способи конструювання set. Особливості використання map.	Лекція	2	
	Вивчення функціональності stack<T>, queue<T>, priority_queue<T>. Використання у власних програмах. Застосування різних реалізацій, дослідження функціонування.	Лабораторна робота	2	Наступне лабораторне заняття
	Всього	Лекцій Лаб занять	28 28	

Обсяг курсу	1 курс 2 семестр Загальний обсяг: 150 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 86 год. – 5 кредитів
	2 курс 1 семестр Загальний обсяг: 120 годин. Аудиторних занять: 42 год., з них 14 год. лекційних та 28 годин лабораторних робіт. Самостійної роботи: 78 год. – 4 кредити
	Загальний обсяг: 270 годин. Аудиторних занять: 106 год., з них 46 год. лекційних та 60 годин лабораторних робіт. Самостійної роботи: 164 год.

Формування програмних компетентностей

Індекс в матриці ОПП	Програмні компоненти
ЗК 2	Здатність застосовувати знання у практичних ситуаціях.
СК 3	Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.
СК 8	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.
ПРН 5	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.
ПРН 9	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.

Літературні джерела

Базові

1. Ярошко С. А. Методи розробки алгоритмів. Програмування мовою C++: навч. посібник / С.А. Ярошко, О.С. Ярошко – Львів: ЛНУ імені Івана Франка, 2022. – 248 с. – <https://lnuittutor.github.io/>
2. Бублик В.В. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТ-книга, 2015. – 624 с.
3. Stephen Prata C++ Primer Plus 6th Edition (Developer's Library) In 2 Volumes – Addison-Wesley Professional, 2011. – 1440 p.

Допоміжні

1. Wentworth P., Elkner J., Downey A., Meyers C. How to Think Like a Computer Scientist: Learning with Python 3. – Green Tea Press, 2018. – 360 p.
2. Sweigart A. Automate the Boring Stuff with Python: Practical Programming for Total Beginners. – No Starch Press, 2014. – 479 p.
3. Bader D. Python Tricks: The Book. — Dan Bader, 2017. – 302 p.
4. SciPy. – <http://scipy.org> .
5. Scikit-Learn. – <http://scikit-learn.org> .

6. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 2nd Edition. – O'Reilly Media, 2018.
7. Langtangen H.P. Texts in Computational Science and Engineering. – Springer-Verlag 2016.

Інформаційні ресурси в Інтернеті

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. Віртуальне навчальне середовище ЛНУП. URL: <https://moodle.lnup.edu.ua/>

Політика оцінювання

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

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

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 ведучим викладачем курсу.

Оцінювання

Остаточна оцінка за курс розраховується наступним чином: поточний контроль оцінюється в 100 балів, та складається із двох модулів по 50 балів кожен. В суму балів кожного модуля входять бали за підготовку, виконання та захисту 10 практичних робіт по 8 балів за кожну роботу ($10 \times 8 = 40$) та 1 бал за самостійну роботу, яка оцінюється усна компонента під час здачі модуля (співбесіда із лектором) ($20 \times 1 = 20$).

Курс 1 семестр 2				Сума	
Поточне тестування та самостійна робота (разом 100 балів)					
Модуль 1 (50 балів)		Модуль 2 (50 балів)		залік	
П1- П5	СР	П6- П10	СР		
5 x 8 =40	10	5 x 8 =20	10		100

Остаточна оцінка за курс розраховується наступним чином: поточний контроль оцінюється в 50 балів, та складається із двох модулів по 25 балів кожен. В суму балів кожного модуля входять бали за підготовку, виконання та захисту 10 практичних робіт по 4 бали за кожну роботу ($10 \times 4 = 40$) та 1 бал за самостійну роботу, яка оцінюється усна компонента під час здачі модуля (співбесіда із лектором) ($10 \times 1 = 10$).

Курс 2 семестр 1 Поточне тестування та самостійна робота (разом 50 балів)				Підсумк овий контрол ь	Сума
Модуль 1 (25 балів)		Модуль 2 (25 балів)		іспит	
П1- П5	СР	П6- П10	СР		
5 x 4 =20	5	5 x 4 =20	5	50	100

П1, П2 ... П10 – практичні роботи; СР – самостійна робота.

До Силабусу також готуються матеріали навчально-методичного комплексу:

- 1) Навчальний контент (розширений план лекцій);
- 2) Тематика та зміст практичних робіт;
- 3) Тематика та методичні рекомендації до виконання курсової роботи;
- 4) Завдання для підсумкової роботи, питання на залік;
- 5) Електронне навчання у віртуальному навчальному середовищі ЛНУП (<https://moodle.lnup.edu.ua/>).