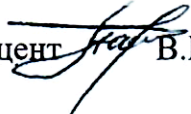


Міністерство освіти і науки України
Львівський національний університет ветеринарної медицини та
біотехнологій імені С.З. Гжицького
Факультет механіки, енергетики та інформаційних технологій
Кафедра інформаційних технологій



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 F3 «Комп'ютерні науки»
перший (бакалаврський) рівень вищої освіти

ВИКЛАДАЧ

Татомир Андрій Володимирович



Електронна пошта: *andrew.tatomyr@gmail.com*

Телефон *+380931128177*

Доцент кафедри інформаційних технологій Львівського національного університету ветеринарної медицини та біотехнологій імені С.З. Гжицького, Staff engineer компанії «Redocly», кандидат технічних наук. Викладач з 17-річним досвідом, автор та співавтор понад 55 наукових статей, 2 – підручників та навчальних посібників, 2 монографій, 1 патента України на винаходи і корисні моделі, 18 навчально-методичних розробок.

Читає курси: Об'єктно-орієнтоване програмування, Основи проектування інформаційних систем, Технології проектування інформаційних систем. Сфера наукових інтересів: програмна інженерія, проектування інтелектуальних інформаційних систем.

ЛЬВІВ 2025

Галузь знань _F «Інформаційні технології»_
 Спеціальність _F3 «Комп'ютерні науки»_
 Освітньо-професійна програма «Комп'ютерні науки»
 Рівень вищої освіти – перший (бакалаврський)
 Кількість кредитів – 8+1
 Рік підготовки, семестр – 1 курс, 2 семестр; 2 курс, 3 семестр
 Компонент освітньої програми: обов'язкова
 Мова викладання: українська

Опис дисципліни

Освітня компонента "Програмування" для здобувачів освітньо-професійної програми «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти охоплює основи та поглиблені аспекти програмування мовою C++, яка є однією з найпотужніших і найпоширеніших мов у сфері розробки програмного забезпечення. Курс розпочинається з вивчення базового синтаксису та структурного програмування, поступово переходячи до об'єктно-орієнтованого підходу (ООП), роботи з пам'яттю, обробки винятків, стандартної бібліотеки шаблонів (STL) та інших сучасних методів програмування. Спрямована на формування у студентів практичних навичок у розробці програм, ефективному використанні ресурсів комп'ютера та написанні оптимізованого коду. Вивчення C++ дозволяє закласти фундамент для подальшого освоєння інших мов програмування, алгоритмів, розробки високопродуктивних додатків і системного програмування.

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 – «Вища математика», «Алгоритми та структури даних», «Архітектура комп'ютерних систем», «Іноземна мова».

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

Предметом вивчення освітньої компоненти «Програмування» є процес навчання і підготовки фахівця за освітньо-професійною програмою «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти, який дозволить застосовувати знання із принципів, методів та засобів розробки програмного забезпечення мовою програмування C++. У межах курсу студенти знайомляться з основними конструкціями мови, вивчають синтаксис, роботу зі змінними, типами даних та операторами. Окрему увагу приділено структурному та процедурному програмуванню, що включає використання функцій, циклів та умовних операторів, роботі з пам'яттю, використання вказівників, динамічному виділенні та звільненні пам'яті, обробці винятків, що дозволяє створювати надійні та безпечні програми. Окрім цього, студенти знайомляться зі стандартною бібліотекою шаблонів (STL), яка містить потужний набір контейнерів, алгоритмів та ітераторів для роботи з даними. Важливим аспектом є також опрацювання файлів, введення та виведення інформації, що дозволяє створювати програми для обробки великих обсягів даних.

Мета навчальної дисципліни. Метою навчальної дисципліни є теоретична і практична підготовка здобувачів вищої освіти, які володіють основними прийомами використання мови програмування високого рівня в інтегрованому середовищі розробки Visual Studio 2022, а також набуття навичок розв'язку типових задач з використанням прикладного програмного забезпечення та сучасних персональних комп'ютерів.

Основними завданнями освітньої компоненти «Програмування» є: формування у студентів фундаментальних знань і практичних навичок роботи з мовою C++. Курс спрямований на засвоєння основ програмування, включаючи розуміння синтаксису, типів даних, операторів та основних конструкцій управління потоком виконання програми, а також розробці структурованих та модульних програм із використанням функцій. Формування навичок роботи з пам'яттю, зокрема використання вказівників та динамічного виділення пам'яті, використання

методів обробки винятків, знайомство з можливостями стандартної бібліотеки шаблонів (STL), робота з файлами та обробка інформації.

Вивчення цієї освітньої компоненти формує у студентів практичні навички розробки ефективних, оптимізованих та надійних програм, що стане міцною основою для їхньої подальшої професійної діяльності у сфері програмної інженерії.

Структура курсу

Години аудиторних занять (лек./ практ.)	Тема	Результати навчання	Завдання
Семестр 2			
2/4	Тема 1. Основи програмування.	Знати що таке комп'ютер, алгоритм, програма. Що він вміє робити Перша програма на C++. Етапи опрацювання програми	Питання, лабораторна робота
2/4	Тема 2. Система типів мови програмування.	Знати структуру програми мовою C++, include, main(). Стандартні заголовкові файли, порядок трансляції. 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 (cmath). Задача про трикутник. Використання cin, cout.	Питання, лабораторна робота
2/4	Тема 3. Інструкції мови C++.	Знати послідовні та галужені алгоритми. Інструкції if, if else, switch, break. Задача min(a,b,c). Повторювані алгоритми. Інструкції циклу for, while, do while. Оголошення в інструкціях. Оператор ?:. Задача про кількість цифр числа. Введення послідовностей чисел (стан потоку введення). Інструкції break, continue, goto. Модифікатор const, тип перелік.	Питання, лабораторна робота
2/4	Тема 4. Оголошення та використання масивів.	Знати модифікатор const, іменовані константи, тип перелік. Загальна структура оголошення імені, typedef, using. Структуровані типи даних: масиви, рядки. Використання масивів для зберігання/побудови послідовності значень. Введення-виведення рядків. Функції для роботи з рядками (cstring).	Питання, лабораторна робота
2/4	Тема 5. Вказівники.	Знати оголошення вказівника, ініціалізація, використання. Створення, використання та знищення динамічних змінних простих типів. Зв'язок вказівників і масивів. Арифметика вказівників. Створення та використання динамічних масивів. Робота з двохвимірними масивами. Побудова лінійного списку за допомогою вказівників.	Питання, лабораторна робота

2/4	Тема 6. Функції.	Знати функції в C++: прототип, визначення, виклик. Модульне програмування. Способи передавання аргументів у функцію. Параметри-значення. Параметри-вказівники. Тип посилання, використання в параметрах функцій.	Питання, лабораторна робота
2/4	Тема 7. Можливості із застосуванням функцій.	Знати функції і масиви (вказівник + розмір, діапазон, робота з частиною масиву). Вказівники, посилання і специфікатор const. Вказівник на функцію (typedef, using, два «імені» однієї функції). Функції вищих порядків (табуляція, застосування до кожного, числові методи). Використання лямбда-виразів. Масиви функцій (текстове меню).	Питання, лабораторна робота
2/4	Тема 8. Рядки. Строки символів.	Знати строка символів – масив Функції для роботи з строками Обробка рядків символів Типізовані показники	Питання, лабораторна робота
Семестр 3			
2/2	Тема 9. Структури.	Знати оголошення структури, поля даних, конструктори. Створення і використання екземплярів структури. Селектор імені. Перевантаження оператора виведення структури. Створення і використання масивів структур. Функції для опрацювання структур. Перевантаження арифметичних операторів.	Питання, лабораторна робота
2/2	Тема 10. Списки.	Знати поняття зв'язної структури даних. Тип для моделювання ланок лінійного однозв'язного списку. Побудова списку з послідовності значень. Перебір елементів списку, виведення на консоль. Вставлення значення у впорядкований список. Вилучення списку з динамічної пам'яті. Сортування списком. Алгоритм злиття впорядкованих послідовностей.	Питання, лабораторна робота
2/2	Тема 11. Деревя.	Знати сортування списком (стосується попередньої лекції). Структура даних дерево. Означення. Поняття рекурсії. Рекурсивні розв'язки, рекурсивні функції. Алгоритми обходу двійкового дерева. Деревя пошуку. Сортування деревом. Збалансовані деревя, вставка, вилучення.	Питання, лабораторна робота
2/2	Тема 12. Потоки введення-виведення даних. Використання файлів.	Знати потоки даних. Виведення в cout. Введення з cin. Введення-виведення файлів	Питання, лабораторна робота
2/2	Тема 13. Особливі випадки оголошення	Знати перевантажені функції. Статичний поліморфізм. Аргументи за замовчуванням функцій. Вбудовані функції.	Питання, лабораторна робота

	функцій. Простори імен. Класи пам'яті.	Знати функції зі змінною (невідомою) кількістю аргументів. Статичні локальні змінні функцій. Простори імен: призначення, оголошення, використання. Класи пам'яті, діапазони доступу, зв'язність імен.	
2/2	Тема 14. Класи.	Знати навіщо класи мови C++. Термінологія. Синтаксис класу. Оголошення полів, методів. Прихована та доступна частини. Розбиття визначення класу на файли. Тестування класу. Надсилання повідомлень. Конструктори: за замовчуванням, створення, копіювання. Деструктор. Порівняння об'єктів, використання вказівника this. Статичні та динамічні об'єкти, масиви об'єктів, вказівники на об'єкт. Діапазон видимості класу. Вкладені оголошення типів.	Питання, лабораторна робота
2/2	Тема 15. Опрацювання помилок, Винятки.	Знати традиційні способи опрацювання помилок. Використання винятків (значення вбудованих типів). Оголошення і використання класів винятків. Стандартні класи винятків. Розгортання стека, повторне генерування винятку. Можливі проблеми, породжені опрацюванням винятку. Додаткові можливості використання винятків.	Питання, лабораторна робота
2/2	Тема 16. Узагальнення в програмуванні. Шаблони.	Знати від конкретних даних до довільних, від частини програми до функції, від конкретного типу до довільного. Шаблон функції: оголошення, використання, явне і неявне створення екземпляра. Часткова спеціалізація, перевантаження шаблону функції. Шаблон класу: оголошення, використання, явне і неявне створення. Часткова спеціалізація шаблону класу.	Питання, лабораторна робота
2/2	Тема 17. Огляд бібліотеки STL.	Знати базові поняття: контейнер, ітератор, алгоритм, функтор. Класифікація ітераторів, моделі ітераторів. Класифікація контейнерів, спільні можливості.	Питання, лабораторна робота
2/2	Тема 18. Використання шаблонів класів і функцій.	Знати наслідування шаблонів <code>template<template>;</code> <code>template<class>; class<template></code> . Включення шаблонів. Вкладений шаблон. Шаблон – параметр шаблону. Клас з шаблонним методом. Шаблон і дружні функції.	Питання, лабораторна робота
2/2	Тема 19. Послідовні контейнери STL.	Знати спільні риси та специфічні можливості. Особливості внутрішнього влаштування Загальні способи використання послідовних контейнерів. Допоміжний «говіркий» клас. Порівняння ефективності. Спеціальні операції зі списком.	Питання, лабораторна робота

2/2	Тема 20. Стек, черга, пріоритетна черга.	Знати особливості реалізації та інтерфейсу. Приклади використання: обхід дерева, польський запис, моделювання обслуговування. Доступ до реалізації. Експерименти з пріоритетною чергою. Власна реалізація контейнерного адаптера. Наслідування від стандартного контейнера для вдосконалення поведінки.	Питання, лабораторна робота
4/4	Тема 21. Алгоритми.	Знати концепцію функтор та її реалізація. Об'єкт-функція. Стандартні функтори, композиція функторів. Каталог алгоритмів. Приклади використання алгоритмів.	Питання, лабораторна робота

Курсова робота

Мета курсової роботи полягає у закріпленні та розширенні теоретичних знань, отриманих під час вивчення дисципліни «Програмування», розвитку навичок самостійного проектування та розробки програм; оволодінні методами та інструментами розробки програмного забезпечення, вмінні застосовувати отримані знання для вирішення конкретних практичних завдань та розвитку вміння оформлення науково-дослідницької роботи.

Завдання курсової роботи з дисципліни «Програмування» включають:

- вибір та обґрунтування теми дослідження;
- огляд та аналіз літератури з обраної теми;
- формулювання мети та завдань дослідження;
- вибір та застосування інструментів розробки для вирішення поставлених завдань;
- отримання та інтерпретація результатів дослідження;
- формулювання висновків та рекомендацій;
- оформлення курсової роботи відповідно до встановлених вимог.

Розподіл балів за виконання курсової роботи

Пояснювальна записка	Вибір методів та інструментів розробки для поставлених завдань	Інтерпретація отриманих результатів та формулювання обґрунтованих висновків	Презентація результатів дослідження та відповіді на запитання	Сума
до 20 балів	до 25 балів	до 35 балів	до 20 балів	100

Орієнтовні Темі курсових робіт, завдання.

1. Розробка прикладної програми для порівняльної оцінки методів сортування даних
2. Розробка додатку для процесів управління підприємством.....
3. Інформаційно-аналітична система підтримки прийняття рішень із узгодження складових технологічної системи
4. Розробка інформаційно-аналітичних додатків в середовищі для обробки даних.
5. Розробка інформаційних ботів для підтримки прийняття рішень в сфері ...

Навчальний контент

Формування програмних компетентностей

Індекс в матриці ОПП	Програмні компоненти
ІК	Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.
ЗК7	Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
СКЗ	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.
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.
СК17	Здатність проектувати та реалізовувати програмне забезпечення для вирішення різноманітних прикладних задач інтелектуальної комп'ютеризації у сфері природокористування із використанням різних мов програмування, сучасних бібліотек крос-платформного програмування, проводити тестування на різних апаратних платформах, впроваджувати і підтримувати роботу інформаційних систем на сучасних платформах.
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.

Літературні джерела

1. Бублик В.В. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. К.: ІТ-книга, 2015. 624 с.
2. Васильєв О.М. Програмування мовою Python. Видавництво: "Навчальна книга - Богдан", Тернопіль, 2019. 504 с.
3. Васильєв О. Програмування на С++ в прикладах і задачах : Навч. посіб. / О. Васильєв. Київ : Видавництво Ліра-К, 2017. 382 с.
4. Каплун В. А. Технологія програмування. Лабораторний практикум : навчальний посібник / В.А. Каплун, Ю.В. Баришев, А.В. Остапенко. Вінниця: ВНТУ, 2015. 125 с.
5. Ментинський С.М., Пелех Я.М. Збірник задач з основ алгоритмізації та програмування. Навчальний посібник з курсів «Обчислювальна техніка та програмування», «Інформатика», «Основи інформатики і програмування», для студентів технічних спеціальностей для першого (бакалаврського) рівня освіти / С. М. Ментинський, Я. М. Пелех. Львів: Галицька Видавнича Спілка, 2023. 320 с.
6. Основи програмування мовою С++ / Путянін Є.П., Степанов В.П., Пчелінов В.П., Долженкова Т.Г., Матат О.О. // Харків.: «Компанія СМІТ». 2015. 319 с.

7. Ярошко С. А. Методи розробки алгоритмів. Програмування мовою C++: навч. посібник / С.А. Ярошко, О.С. Ярошко. Львів: ЛНУ ім. І.Франка, 2022. 248 с. <https://lnuittutor.github.io/>
8. Bruce Eckel. Thinking in C++, Vol. 1: Introduction to Standard C++, 2nd Edition. Prentice Hall; (March 25, 2020), 840 p.
9. Stephen Prata. C++ Primer Plus (Developer's Library). Addison-Wesley Professional; 6th edition (October 18, 2021). 1440 p.

Допоміжна

10. Глинський Я.М. C++ і C++ Builder / Глинський Я.М., Анохін В.С., Ряжська В.А. // Львів: Деол, СПД Глинський, 2013. 192 с.

Інформаційні ресурси

1. Бібліотечно-інформаційні ресурси – книжковий фонд, періодика та фонди на електронних носіях бібліотеки ЛНУВМБ, 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
3. Задачі програмування із прикладами розв'язку: <http://purecodecpp.com/uk/archives/433>
4. Задачі програмування із прикладами розв'язку: <http://library.nuft.edu.ua/ebook/datathree.php?ID=138>
5. Задачі програмування із прикладами розв'язку: <http://abramov.org.ua/blog/category/opp/obchislennya-%D1%96z-zber%D1%96gannyam-posl%D1%96dovnosti/>
6. ІТ компанії Львова: http://it-catalogue.net/ru/component/companies_cat/companies/458/all/default/all/main.html
7. Книжки з програмування: як читати і що саме: <https://dou.ua/lenta/articles/programming-books/>
8. Код тестових програм. URL: github.com/krenevych/algo
9. Українська технічна література. Програмування: <https://ukrtechlibrary.wordpress.com/tag/програмування/>
10. Портал об'єктно-орієнтованого програмування: <http://oop.in.ua/tag/FAQ/>
11. Eolymp. URL: eolymp.com

Політика оцінювання

Успішність студента оцінюється шляхом проведення поточного та підсумкового контролю.

Максимальна кількість балів з дисципліни «Програмування», яку може отримати студент протягом семестру за всі види роботи, становить 100, при цьому 50 балів за результатами поточного оцінювання, та 50 – за результатами екзаменаційного контролю.

Результати поточного контролю оцінюються за чотирибальною («2», «3», «4», «5») шкалою. В кінці семестру обчислюється середнє арифметичне значення (САЗ) усіх отриманих студентом оцінок з наступним переведенням його у 50-ти бальну шкалу за формулою:

$$ПК = 10 \cdot САЗ$$

Критерії поточного оцінювання знань студентів

Оцінка	Критерії оцінювання
5 («відмінно»)	У повному обсязі володіє навчальним матеріалом, вільно, самостійно та аргументовано його викладає, глибоко і всебічно розкриває зміст, використовуючи обов'язкову та додаткову літературу. Правильно вирішив 90% тестових завдань.

4 («добре»)	Достатньо повно володіє навчальним матеріалом, обґрунтовано його викладає, в основному розкриває зміст завдань, використовуючи обов'язкову літературу. При викладанні окремих питань не вистачає достатньої глибини та аргументації, допускаються несуттєві неточності й незначні помилки. Правильно вирішив більшість тестових завдань.
3 («задовільно»)	У цілому володіє навчальним матеріалом, викладає його основний зміст, але без глибокого всебічного аналізу, обґрунтування та аргументації, допускаючи окремі суттєві неточності та помилки. Правильно вирішив близько половини тестових завдань.
2 («незадовільно»)	Не в повному обсязі володіє навчальним матеріалом. Викладає матеріал фрагментарно та поверхово, без аргументації й обґрунтування, недостатньо розкриває зміст теоретичних і практичних завдань, допускає суттєві неточності. Правильно вирішив меншість тестових завдань.

Переведення підсумкових рейтингових оцінок з дисципліни, виражених у балах за 100-бальною шкалою, у оцінки за національною шкалою та шкалою ECTS.

Таблиця 1 – Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, диференційованого заліку, курсового проєкту (роботи), практики	для заліку
90–100	A	відмінно	зараховано
82–89	B	добре	
74–81	C		
64–73	D	задовільно	
60–63	E		
35–59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0–34	F	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

Розподіл балів для дисциплін, які завершуються ЗАЛІКОМ.

Для здобувачів вищої освіти денної форми навчання

Максимальна кількість балів за засвоєння тем дисципліни протягом семестру становить 100:

$$100(PK) = 100$$

де: 100 (ПК) 100 максимальних балів з поточного контролю, які може набрати Здобувач за семестр.

$$PK = \frac{100 \cdot CAZ}{5} = 20 \cdot CAZ$$

Для Здобувачів вищої освіти заочної форми навчання

$$30(PK) + 70(CP) = 100$$

30 (ПК) – 30 максимальних балів з поточного контролю (ПК), які може набрати здобувач під час настановної та лабораторно-екзаменаційної сесії.

70 (ТСР) бали за виконання тематичної самостійної роботи у міжсесійний період за програмою курсу.

За підсумками семестрового контролю в залікову відомість здобувачу у графі «за національною шкалою» виставляється оцінка «зараховано/не зараховано».

Максимальна кількість балів за курсову роботу (проект) становить 100. Критерії оцінювання курсових робіт (проектів) представлено у таблиці 1.

Захист курсових робіт (проектів) здійснюється перед комісією у складі 2-3 викладачів кафедри, у тому числі - керівника курсової роботи (проекту).

До Силабусу також готуються матеріали навчально-методичного комплексу:

- 1) Навчальний контент;
- 2) Тематика та зміст лабораторних робіт;
- 3) Завдання для підсумкової роботи, питання на іспит;
- 4) Електронне навчання у віртуальному навчальному середовищі ЛНУВМБ (<https://moodle.lnup.edu.ua/>).