

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА
БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ГІЖИЦЬКОГО

Факультет механіки, енергетики та інформаційних технологій
Кафедра інформаційних технологій

ПОГОДЖЕНО

Гарант ОПП «Комп'ютерні науки»
Вадим ПТАШНИК

(ім'я та прізвище, підпис)

«27» серпня 2025 року

ЗАТВЕРДЖЕНО

Декан факультету механіки, енергетики та
інформаційних технологій
Степан КОВАЛИШИН

(ім'я та прізвище, підпис)

«28» серпня 2025 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ПРОГРАМУВАННЯ»

(назва навчальної дисципліни)

рівень вищої освіти	<u>перший (бакалаврський)</u> (назва освітнього рівня)
галузь знань	<u>F «Інформаційні технології»</u> (назва галузі знань)
спеціальність	<u>F3 «Комп'ютерні науки»</u> (назва спеціальності)
освітня програма	<u>«Комп'ютерні науки»</u> (назва)
вид дисципліни	<u>обов'язкова</u> (обов'язкова / за вибором)
програма навчання	<u>повна</u> (повна/ скорочена)

2025-2026 навчальний рік

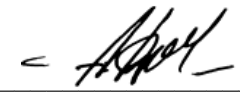
Робоча програма «Програмування»
(назва навчальної дисципліни)

Укладачі: Татомир А.В. – к.т.н., доцент кафедри інформаційних технологій, Заплатинський Н.Б. –
ст.викладач.

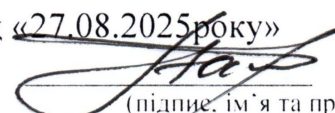
(вказати укладачів, їхні посади, наукові ступені та вчені звання)

Робочу програму схвалено на засіданні кафедри інформаційних технологій.

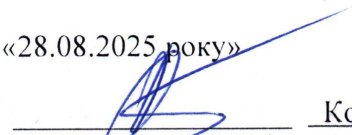
Протокол № 1 від 26.08.2025 року.

Завідувач кафедри  Тригуба А.М.
(підпис, ім'я та прізвище)

Погоджено навчально-методичною комісією спеціальності
«Комп'ютерні науки»
(назва спеціальності)

Протокол № 8/1 від «27.08.2025 року»
Голова НМКС  Пташник В.В.
(підпис, ім'я та прізвище)

Схвалено рішенням навчально-методичної ради факультету ФМЕІТ
(назва факультету)

Протокол № 1 від «28.08.2025 року»
Голова НМРФ  Ковалишин С.Й.
(підпис, ім'я та прізвище)

Ухвалено вченою радою ФМЕІТ протокол №1 від 28.08.2025 р.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Всього годин	
	денна форма здобуття освіти	заочна форма здобуття освіти
Семестр	2-3	2-3
Кількість кредитів/годин	9/270	9/270
Усього годин аудиторної роботи	104	46
в т.ч.:		
• лекційні заняття, год.	44	22
• практичні заняття, год.	–	–
• лабораторні заняття, год.	60	24
• семінарські заняття, год.	–	–
Усього годин самостійної роботи	166	224
Форма контролю	іспит, іспит, курсова робота	іспит, іспит, курсова робота

Примітка.

Частка аудиторного навчального часу студента відносно годин самостійної роботи у відсотковому вимірі:

для денної форми здобуття освіти – 38,5%;

для заочної форми здобуття освіти – 17%.

1. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Метою вивчення освітньої компоненти «Програмування» 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.

Завдання навчальної дисципліни передбачають: 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.

Преквізити: для успішного опанування курсу «Програмування» необхідно володіти знаннями з інформатики та іноземної (англійської) мови, вищої математики, математичного аналізу, чисельних методів, архітектури комп'ютерних систем, алгоритмів та структур даних, які формуються під час отримання загальної освіти у школі, а також вивчення дисциплін «Вища математика», «Математичний аналіз», «Чисельні методи», «Архітектура комп'ютерних систем», «Алгоритми та структури даних».

Постреквізити: вивчення дисципліни «Програмування» створює підґрунтя для опанування наступних компонент освітньої програми. Здобуті знання безпосередньо використовуються у курсах «Об'єктно-орієнтоване програмування», «Вебтехнології», «Бази даних», «Якість програмного забезпечення та тестування», «Моделювання систем», «Інформаційна безпека», «Клієнт-серверне програмування» та «Основи штучного інтелекту», де вони застосовуються для створення програмних рішень, побудови ефективних інформаційних систем, забезпечення їх надійності, безпеки та оптимальної взаємодії між компонентами тощо.

Важливими постреквізитами є «Виробнича практика» та «Передкваліфікаційна практика», під час яких набуті компетентності реалізуються у реальних умовах організації робочих процесів та взаємодії в командах.

Результати навчання за дисципліною «Програмування» зумовлюють навички щодо застосування знань із принципів, методів та засобів розробки програмного забезпечення. 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.

Відповідно до освітньо-професійної програми «Комп'ютерні науки» вивчення дисципліни забезпечує набуття здобувачами таких компетентностей та програмних результатів навчання:

Загальні компетентності	ЗК7 Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
Фахові (спеціальні) компетентності	- СК3 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.
Програмні результати навчання	- 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.

2. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви тем	Кількість годин											
	денна форма здобуття освіти (ДФЗО)						заочна форма здобуття освіти (ЗФЗО)					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	ін д.	с. р.		л	п	лаб	ін д.	с. р.
1	2	3	4	5	6	7	8	9	10	11	12	13
	Рік підготовки 1 Семестр 2						Рік підготовки 1 Семестр 2					
Тема 1. Основи програмування.	12	2		4		6	12	2				10
Тема 2. Система типів мови програмування.	10	2		4		4	10	2				8
Тема 3. Інструкції мови C++.	12	2		4		6	12	2		2		8
Тема 4. Оголошення та використання масивів.	10	2		4		4	10	-		2		8
Тема 5. Вказівники.	12	2		4		6	12	2		2		8
Тема 6. Функції.	10	2		4		4	10	1		2		7
Тема 7. Можливості із застосуванням функцій.	12	2		4		6	12	1		2		9
Тема 8. Рядки. Строки символів.	12	2		4		6	12	1		2		9
<i>Підготовка до навчальних занять та контрольних заходів.</i>	30	-	-	-	-	30	30	-	-	-	-	30
Разом за семестр 2	120	16	0	32	0	72	120	11	0	12	0	97
	Рік підготовки 2 Семестр 3						Рік підготовки 2 Семестр 3					
Тема 9. Структури.	6	2		2		2	6			2		4
Тема 10. Списки.	6	2		2		2	6	2				4
Тема 11. Дерева.	6	2		2		2	6			2		4
Тема 12. Потоки введення-виведення даних. Використання файлів.	6	2		2		2	6	1				5
Тема 13. Особливі випадки оголошення функцій. Простори імен. Класи пам'яті.	6	2		2		2	6			2		4
Тема 14. Класи.	6	2		2		2	6	1				5
Тема 15. Опрацювання помилок, Винятки.	6	2		2		2	6	1				5
Тема 16. Узагальнення в програмуванні. Шаблони.	6	2		2		2	6			2		4
Тема 17. Огляд бібліотеки STL.	6	2		2		2	6	1				5
Тема 18. Використання шаблонів класів і функцій.	8	2		2		4	8	1		2		5
Тема 19. Послідовні контейнери STL.	8	2		2		4	8	2				6
Тема 20. Стек, черга, пріоритетна черга.	8	2		2		4	8	2				6
Тема 21. Алгоритми.	12	4		4		4	12			2		10

Підготовка до навчальних занять та контрольних заходів.	30	–	–	–	–	30	30	–	–	–	–	30
Разом за семестр 3	120	28	0	28	0	64	120	11	0	12	0	97
Індивідуальні завдання												
КР	30	–	–	–	–	30	30	–	–	–	–	30
Усього годин	270	44	–	60	0	166	270	22	–	24	0	224

3. ЛЕКЦІЙНІ ЗАНЯТТЯ

№ з/п	Назви тем та їх короткий зміст	Кількість годин	
		ДФЗО	ЗФЗО
1	Тема 1. Основи програмування. 1. Що таке комп'ютер, алгоритм, програма. 2. Що він вміє робити Перша програма на C++. 3. Етапи опрацювання програми	2	2
2	Тема 2. Система типів мови програмування. 1. Структура програми мовою C++, include, main(). Стандартні заголовкові файли, порядок трансляції. Система типів. 2. Логічний тип: зображення, оператори, вирази. 3. Літерний тип. Зображення літер. 4. Цілі типи: назви, граничні значення, зображення, оператори. Оператори присвоєння. 5. Дійсні типи. Перетворення типів. Математичні функції (cmath). Задача про трикутник. 6. Використання cin, cout.	2	2
3	Тема 3. Інструкції мови C++. 1. Послідовні та галужені алгоритми. 2. Інструкції if, if else, switch, break. Задача min(a,b,c). 3. Повторювані алгоритми. Інструкції циклу for, while, do while. 4. Оголошення в інструкціях. 5. Оператор ?:. Задача про кількість цифр числа. 6. Введення послідовностей чисел (стан потоку введення). Інструкції break, continue, goto. 7. Модифікатор const, тип перелік.	2	2
4	Тема 4. Оголошення та використання масивів. 1. Модифікатор const, іменовані константи, тип перелік. 2. Загальна структура оголошення імені, typedef, using. 3. Структуровані типи даних: масиви, рядки. 4. Використання масивів для зберігання/побудови послідовності значень. 5. Введення-виведення рядків. Функції для роботи з рядками (cstring).	2	-
5	Тема 5. Вказівники. 1. Оголошення вказівника, ініціалізація, використання. 2. Створення, використання та знищення динамічних змінних простих типів. 3. Зв'язок вказівників і масивів. Арифметика вказівників. 4. Створення та використання динамічних масивів. 5. Робота з двохвимірними масивами. 6. Побудова лінійного списку за допомогою вказівників.	2	2
6	Тема 6. Функції. 1. Функції в C++: прототип, визначення, виклик. 2. Модульне програмування. 3. Способи передавання аргументів у функцію. 4. Параметри-значення. 5. Параметри-вказівники.	2	1

	6. Тип посилання, використання в параметрах функцій.		
7	Тема 7. Можливості із застосуванням функцій. 1. Функції і масиви (вказівник + розмір, діапазон, робота з частиною масиву). 2. Вказівники, посилання і специфікатор const. 3. Вказівник на функцію (typedef, using, два «імені» однієї функції). 4. Функції вищих порядків (табуляція, застосування до кожного, числові методи). 5. Використання лямбда-виразів. 6. Масиви функцій (текстове меню).	2	1
8	Тема 8. Рядки. Строки символів. 1. Строка символів – масив 2. Функції для роботи з строками 3. Обробка рядків символів 4. Типізовані покажчики	2	1
9	Тема 9. Структури. 1. Оголошення структури, поля даних, конструктори. 2. Створення і використання екземплярів структури. Селектор імені. 3. Перевантаження оператора виведення структури. 4. Створення і використання масивів структур. 5. Функції для опрацювання структур. 6. Перевантаження арифметичних операторів.	2	
10	Тема 10. Списки. 1. Поняття зв'язної структури даних. 2. Тип для моделювання ланок лінійного однозв'язного списку. 3. Побудова списку з послідовності значень. 4. Перебір елементів списку, виведення на консоль. 5. Вставляння значення у впорядкований список. 6. Вилучення списку з динамічної пам'яті. 7. Сортування списком. Алгоритм злиття впорядкованих послідовностей.	2	2
11	Тема 11. Деревя. 1. Сортування списком (стосується попередньої лекції). 2. Структура даних дерево. Означення. 3. Поняття рекурсії. Рекурсивні розв'язки, рекурсивні функції. 4. Алгоритми обходу двійкового дерева. 5. Деревя пошуку. Сортування деревом. 6. Збалансовані деревя, вставка, вилучення.	2	
12	Тема 12. Потоки введення-виведення даних. Використання файлів. 1. Потоки даних. 2. Виведення в cout. 3. Введення з cin. 4. Введення-виведення файлів.	2	1
13	Тема 13. Особливі випадки оголошення функцій. Простори імен. Класи пам'яті. 1. Перевантажені функції. Статичний поліморфізм. 2. Аргументи за замовчуванням функцій. 3. Вбудовані функції. 4. Функції зі змінною (невідомою) кількістю аргументів. 5. Статичні локальні змінні функцій. 6. Простори імен: призначення, оголошення, використання. 7. Класи пам'яті, діапазони доступу, зв'язність імен.	2	
14	Тема 14. Класи. 1. Навіщо класи мові C++. Термінологія. 2. Синтаксис класу. Оголошення полів, методів. Прихована та доступна частини.	2	1

	3. Розбиття визначення класу на файли. Тестування класу. Надсилання повідомлень. 4. Конструктори: за замовчуванням, створення, копіювання. Деструктор. 5. Порівняння об'єктів, використання вказівника this. 6. Статичні та динамічні об'єкти, масиви об'єктів, вказівники на об'єкт. 7. Діапазон видимості класу. Вкладені оголошення типів.		
15	Тема 15. Опрацювання помилок, Винятки. 1. Традиційні способи опрацювання помилок. 2. Використання винятків (значення вбудованих типів). 3. Оголошення і використання класів винятків. 4. Стандартні класи винятків. 5. Розгортання стека, повторне генерування винятку. 6. Можливі проблеми, породжені опрацюванням винятку. 7. Додаткові можливості використання винятків.	2	1
16	Тема 16. Узагальнення в програмуванні. Шаблони. 1. Від конкретних даних до довільних, від частини програми до функції, від конкретного типу до довільного. 2. Шаблон функції: оголошення, використання, явне і неявне створення екземпляра. 3. Часткова спеціалізація, перевантаження шаблону функції. 4. Шаблон класу: оголошення, використання, явне і неявне створення. 5. Часткова спеціалізація шаблону класу.	2	
17	Тема 17. Огляд бібліотеки STL. 1. Базові поняття: контейнер, ітератор, алгоритм, функтор. 2. Класифікація ітераторів, моделі ітераторів. 3. Класифікація контейнерів, спільні можливості.	2	1
18	Тема 18. Використання шаблонів класів і функцій. 1. Наслідування шаблонів <code>template<template; template<class; class<template.</code> 2. Включення шаблонів. 3. Вкладений шаблон. 4. Шаблон – параметр шаблону. 5. Клас з шаблонним методом. 6. Шаблон і дружні функції.	2	1
19	Тема 19. Послідовні контейнери STL. 1. Спільні риси та специфічні можливості. Особливості внутрішнього влаштування 2. Загальні способи використання послідовних контейнерів. 3. Допоміжний «говіркий» клас. Порівняння ефективності. 4. Спеціальні операції зі списком.	2	2
20	Тема 20. Стек, черга, пріоритетна черга. 1. Особливості реалізації та інтерфейсу. 2. Приклади використання: обхід дерева, польський запис, моделювання обслуговування. 3. Доступ до реалізації. Експерименти з пріоритетною чергою. 4. Власна реалізація контейнерного адаптера. 5. Наслідування від стандартного контейнера для вдосконалення поведінки.	2	2
21	Тема 21. Алгоритми. 1. Концепція функтор та її реалізація. Об'єкт-функція. 2. Стандартні функтори, композиція функторів. 3. Каталог алгоритмів. 4. Приклади використання алгоритмів.	4	
Усього годин		44	22

4. ЛАБОРАТОРНІ ЗАНЯТТЯ

№ з/п	Назви тем та їх короткий зміст	Кількість годин	
		ДФЗО	ЗФЗО
Семестр 2			
1	Робота в інтегрованому середовищі розробки Microsoft Visual Studio	4	1
2	Ввід та вивід інформації в консольному режимі. Потоків та форматовані операції мови C++	4	1
3	Лінійні програми. Обчислення арифметичних виразів та математичних функцій	4	2
4	Оператори розгалуження. Оператор <i>if</i>	4	2
5	Оператори розгалуження. Оператор <i>switch</i>	4	2
6	Робота з елементами одновимірних масивів	4	2
7	Робота з елементами багатовимірних масивів	4	2
8	Прості цикли із відомим числом повторів	4	2
Семестр 3			
9	Прості цикли із невідомим числом повторів	2	1
10	Програмування з використанням функцій	4	1
11	Вкладені цикли та задачі обробки масивів	4	1
12	Використання рекурсії та аналіз її типів	4	1
13	Робота із файлами	2	2
14	Генерування псевдовипадкових чисел	4	2
15	Робота із рядковими та символьними змінними	4	2
16	Програмування з використанням структур	4	2
Усього годин		60	24

5. САМОСТІЙНА РОБОТА

№ з/п	Назви тем та їх короткий зміст	Кількість годин	
		ДФЗО	ЗФЗО
1	Алгоритмічна складність. Поліноміальна та неполіноміальна складність алгоритмів.	4	10
2	Програмне забезпечення ПК. Мови програмування.	4	8
3	Записи, множини, покажчики.	4	8
4	Процедури створення інтерфейсу прикладного застосування.	4	8
5	Засоби інтегрованого середовища розробки програмного коду (вбудований відладчик, оглядач проекту, сховище об'єктів та довідкова система).	4	8
6	Бібліотеки компонентів, базові класи оболонки програмного середовища.	4	7
7	Помилки програмування. Чистий код.	4	9
8	Порядок застосування виключних ситуацій, їх протоколювання. Коди помилок при виникненні виключних ситуацій.	4	9
9	Сучасні тенденції розвитку програмних середовищ.	4	4
10	Робота класів та об'єктів, полів, властивостей, методів, повідомлень та подій.	4	4
11	Розв'язання задач з масивами.	4	4
12	Розв'язання задач з рядками.	4	5
13	Розв'язання задач з даними типу структура.	4	4

14	Розв'язання задач з файлами.	3	5
15	Чистий код. Перевірка чистоти коду.	3	5
16	Переваги об'єктно-орієнтованого програмування.	3	4
17	Базові алгоритми сортування. Сортування дерев даних.	3	5
18	Системи числення та їх різновиди.	3	5
19	Особливості класів – перевантаження операторів.	3	6
20	Особливості класів – перетворення типу. Явні конструктори.	3	6
21	Особливості класів – коли необхідно оголошувати конструктор копіювання і оператор присвоєння.	3	10
Підготовка до навчальних занять та контрольних заходів		60	60
КР		30	30
Усього годин		166	224

6. ІНДИВІДУАЛЬНІ ЗАВДАННЯ

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 ...

7. МЕТОДИ НАВЧАННЯ

Навчання з дисципліни «Програмування» здійснюється із застосуванням сучасних інтерактивних та практикоорієнтованих методів, що поєднують словесні (лекція, пояснення,

дискусія), наочні (демонстрація, робота з мультимедійними матеріалами) та активні форми (групові проекти, семінари-дискусії, моделювання ситуацій, аналіз кейсів).

Використання методів мозкового штурму, проблемно-орієнтованих і дослідницьких підходів сприяє розвитку критичного та креативного мислення, уміння працювати в команді й приймати ефективні управлінські рішення. Ефективність забезпечується залученням сучасних цифрових інструментів, програмних середовищ для створення, відлагодження та редагування програм, а також роботи з професійною літературою та науковими публікаціями.

8. МЕТОДИ КОНТРОЛЮ

Оцінювання результатів навчання студентів здійснюється проведенням поточного та підсумкового контролю.

Поточний контроль здійснюється під час практичних занять і має на меті перевірку рівня підготовленості студента до виконання відповідних завдань. Форми проведення поточного контролю – усне та письмове опитування, тестовий контроль.

Підсумковий контроль проводиться з метою оцінювання результатів навчання на завершальному етапі вивчення дисципліни. Підсумковий контроль здійснюється у формі екзамену.

9. КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ

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:

$$ПК = 10 \cdot \text{САЗ}.$$

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

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

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

10. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Підручники і навчальні посібники; інструктивно-методичні матеріали до практичних занять; індивідуальні навчально-дослідні завдання; контрольні роботи; текстові та електронні варіанти тестів для поточного контролю, методичні матеріали для організації самостійної роботи студентів, виконання індивідуальних завдань.

Посилання на матеріали розміщені в Мудл та в Інтернеті.

11. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Базова

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 C++ 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 C++ / 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., Рязька В.А. // Львів:

Деол, СПД Глинський, 2013. 192 с.

12. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Бібліотечно-інформаційні ресурси

- Бібліотека ЛНУВМБ: м. Дубляни, вул. В.Великого, 1; тел. 22-45-915
- Львівська наукова бібліотека ім. Стефаника НАН України: вул. Стефаника, 2; тел. 74-43-72
- Львівська обласна наукова бібліотека: просп. Шевченка, 13; тел. 74-02-26
- Наукова бібліотека ЛНУ ім. Франка, метод. відділ: вул. Драгоманова, 17; тел. 296-42-41
- Центральна міська бібліотека ім. Лесі Українки: вул. Мулярська, 2а; тел. 72-05-81

2. Інформаційні ресурси в Інтернеті

1. Електронний ресурс стосовно ІТ компаній Львівського регіону:
http://www.invest-lvivregion.com/it-компанії_ua_285cms.htm
2. Задачі програмування із прикладами розв'язку: <http://purecodecpp.com/uk/archives/433>
3. Задачі програмування із прикладами розв'язку:
<http://library.nuft.edu.ua/ebook/datathree.php?ID=138>
4. Задачі програмування із прикладами розв'язку:
<http://abramov.org.ua/blog/category/opp/obchislennya-%D1%96z-zber%D1%96gannyam-posl%D1%96dovnosti/>
5. ІТ компанії Львова:
http://it-catalogue.net/ru/component/companies_cat/companies/458/all/default/all/main.html
6. Книжки з програмування: як читати і що саме:
<https://dou.ua/lenta/articles/programming-books/>
7. Код тестових програм. URL: github.com/krenevych/algo
8. Українська технічна література. Програмування:
<https://ukrtechlibrary.wordpress.com/tag/програмування/>
9. Портал об'єктно-орієнтованого програмування: <http://oop.in.ua/tag/FAQ/>
10. Eolymp. URL: eolymp.com