Чернівецький національний університет імені Юрія Федьковича

Факультет математики та інформатики

Кафедра прикладної математики та інформаційних технологій

“ЗАТВЕРДЖУЮ”
Декан факультету математики та інформатики

_____ Ольга МАРТИНЮК
“ 28 ” серпня 2026 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Об’єктно-орієнтоване програмування

(назва навчальної дисципліни)

Обов’язкова навчальна дисципліна

(вказати: обов’язкова / вибіркова)

Освітньо-професійна програма: Системний аналіз

(назва програми)

Спеціальність: F4 «Системний аналіз та наука про дані»

(вказати: код, назва)

Галузь знань: F – Інформаційні технології

(вказати: шифр, назва)

Рівень вищої освіти: перший (бакалаврський)

(вказати: перший бакалаврський / другий магістерський)

Факультет: факультет математики та інформатики

(назва факультету/інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання: українська

(вказати: на якій мові читається дисципліна)

Чернівці – 2026

Робоча програма навчальної дисципліни «Об’єктно-орієнтоване програмування» складена відповідно до освітньо-професійної програми «Системний аналіз» підготовки здобувачів вищої освіти за спеціальністю F4 Системний аналіз та наука про дані (галузь знань F Інформаційні технології). Дисципліна реалізує обов’язковий освітній компонент ОК19 «Об’єктно-орієнтоване програмування» в обсязі 5,0 кредитів ECTS (150 годин).

Розробник: Тетяна Миколаївна Сопронюк, доцент, канд. фіз.-мат. наук, доцент

(П.І.Б. автора, посада, науковий ступінь, вчене звання)

Викладач, що забезпечує читання даної навчальної дисципліни: Тетяна Миколаївна Сопронюк, доцент, канд. фіз.-мат. наук, доцент

(П.І.Б. викладача, посада, науковий ступінь, вчене звання)

Погоджено з гарантом ОП _____ Андрій ПЕРЦОВ

(підпис)

Затверджено на засіданні кафедри прикладної математики та інформаційних технологій

Протокол № 2 від « 28 » серпня 2026 року

Завідувач кафедри _____ Ярослав БІГУН

(підпис)

Схвалено методичною радою факультету математики та інформатики

Протокол № 1 від « 28 » серпня 2026 року

Голова методичної ради _____ Тоня ФРАТАВЧАН

(підпис)

Мета навчальної дисципліни

Поглиблене оволодіння сучасними технологіями програмування на С++ з використанням класів, успадкування і поліморфізму. Для досягнення мети студенти повинні оволодіти програмним матеріалом, написати програми, виконати модульні контрольні роботи, скласти залік.

Курс присвячено вивченню сучасних технологій програмування на С++ (парадигми процедурного, модульного і об'єктно-орієнтованого програмування) з використанням класів, успадкування і поліморфізму. У курсі демонструється застосування об'єктно-орієнтованого підходу для розробки програмного забезпечення.

Пререквізити: програмування (освітній компонент ОК25 «Програмування»).

Результати навчання. У результаті вивчення навчальної дисципліни студент повинен:

- **знати:** поняття базового і породженого (похідного) класів, ієрархії класів; методику програмування, що концентрує основну увагу на зв'язках між об'єктами, а не на деталях їхньої реалізації (інкапсуляція дає можливість об'єднання даних і алгоритмів їх обробки, в результаті чого і дані, і процедури багато в чому втрачають самостійне значення);
- **вміти:** застосовувати теоретичні знання для створення власних класів, перевантажувати оператори для структур і класів, перевизначати функції, застосовувати шаблони функцій і класів, успадковувати класи, побудувати ієрархію класів, використовувати віртуальні функції і абстрактні класи. При написанні лабораторних робіт і, взагалі, програмних систем студент повинен вміти застосовувати об'єктно-орієнтований підхід.

Під час вивчення дисципліни, відповідно до освітньо-професійної програми «Системний аналіз», формуються такі:

загальні компетентності:

- **ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.**
- **ЗК02. Здатність застосовувати знання у практичних ситуаціях.**
- **ЗК03. Здатність планувати і управляти часом.**
- **ЗК04. Знання та розуміння предметної області та розуміння професійної діяльності.**
- **ЗК05. Здатність спілкуватися державною мовою усно і письмово.**
- **ЗК10. Здатність працювати автономно.**
- **ЗК11. Здатність генерувати нові ідеї (креативність).**
- **ЗК14. Здатність оцінювати та забезпечувати якість виконуваних робіт.**

фахові компетентності:

- **ФК1. Здатність використовувати системний аналіз як сучасну міждисциплінарну методологію, що базується на прикладних математичних методах та сучасних інформаційних технологіях і орієнтована на вирішення задач аналізу і синтезу технічних, економічних, соціальних, екологічних та інших складних систем.**

- **ФК2. Здатність формалізувати проблеми, описані природною мовою, у тому числі за допомогою математичних методів, застосовувати загальні підходи до математичного моделювання конкретних процесів.**
- **ФК6. Здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проєктувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації, обробки інформації, інтелектуального аналізу даних.**
- **ФК7. Здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем, а саме: об'єктно-орієнтований підхід при проєктуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань.**
- **ФК8. Здатність організовувати роботу з аналізу та проєктування складних систем, створення відповідних інформаційних технологій та програмного забезпечення.**
- **ФК9. Здатність представляти математичні аргументи і висновки з них з ясністю і точністю і в таких формах, які підходять для аудиторії як усно, так і в письмовій формі.**
- **ФК10. Здатність розробляти експериментальні та спостережувальні дослідження і аналізувати дані, отримані в них.**

та отримуються такі **програмні результати навчання:**

- **ПР8. Володіти сучасними методами розробки програм і програмних комплексів та прийняття оптимальних рішень щодо складу програмного забезпечення, алгоритмів процедур і операцій.**
- **ПР9. Вміти створювати ефективні алгоритми для обчислювальних задач системного аналізу та систем підтримки прийняття рішень.**
- **ПР12. Застосовувати методи і засоби роботи з даними і знаннями, методи математичного, логіко-семантичного, об'єктного та імітаційного моделювання, технології системного і статистичного аналізу.**
- **ПР13. Проєктувати, реалізовувати, тестувати, впроваджувати, супроводжувати, експлуатувати програмні засоби роботи з даними і знаннями в комп'ютерних системах і мережах.**
- **ПР15. Розуміти українську та іноземну мови на рівні, достатньому для обробки фахових інформаційно-літературних джерел, професійного усного і письмового спілкування, написання текстів за фаховою тематикою.**

Опис навчальної дисципліни

Загальна інформація

Назва навчальної дисципліни: Об'єктно-орієнтоване програмування												
Форма навчання	**Рік підготовки**	**Семестр**	**Кількість кредитів**	**Кількість годин**	**Змістових модулів**	**Лекції**	**Практичні**	**Семінарські**	**Лабораторні**	**Самостійна робота**	**Індивідуальні завдання**	**Вид підсумкового контролю**
Денна	**2**	**3**	5	150	3	30	—	—	30	90	—	**Залік**

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Усього годин (денна форма)	Лекції (л)	Практичні (п)	Лабораторні (лаб)	Інд. завд. (інд)	Самост. робота (с.р.)
1	**2**	**3**	**4**	**5**	**6**	**7**
Змістовий модуль 1. Нові можливості C++, які не пов'язані з ООП						
Тема 1. Парадигма програмування. Процедурне, модульне і об'єктно-орієнтоване програмування. Вступ. Парадигма програмування. Процедурне програмування. Модульне програмування. Об'єктно-орієнтоване програмування. Загальна характеристика мови C++ як мови, що найбільш втілює ідеї об'єктно-орієнтованого програмування.	20	4	—	4	—	12
Тема 2. Розширені можливості мови C++, що не пов'язані з класами. Прототипи функцій, аргументи за замовчуванням, посилання. Модифікатори const і volatile. Функціональний запис перетворення типів. Доступ до глобальних змінних, які закриті локальними. Функції, що підставляються. Оператори динамічного розподілу пам'яті. Перевантаження функцій, шаблони функцій. Перевантаження операторів для структур. Перевантаження операцій потокового вводу і виводу для структур. Приклади перевантаження операцій і функцій, використання шаблонів.	30	6	—	6	—	18
Разом за змістовим модулем 1	**50**	**10**	**—**	**10**	**—**	**30**
Змістовий модуль 2. Інкапсуляція. Поняття класу						
Тема 3. Класи і об'єкти. Створення і знищення об'єктів. Поняття класу та його визначення. Специфікатори public, private. Доступ до відкритих членів класу. Функції доступу до захованих членів класу. Створення об'єктів. Статичні члени класу. Дружні і складові функції. Дружні класи. Ініціалізація та знищення об'єктів. Автоматичне і динамічне виділення пам'яті під об'єкти класу. Конструктори і деструктори.	24	5	—	4	—	15
Тема 4. Операції над класовими об'єктами. Перевизначення операцій для класів. Бінарні і унарні операції. Пріоритети при перевизначенні операцій. Присвоєння і ініціалізації. Перевантаження операторів виклику функції, індексування. Стандартні об'єкти потокового вводу-виведення C++. Перевантаження операцій "<<", ">>" для класу користувача. Приклади проєктування і побудови класів.	26	5	—	6	—	15
Разом за змістовим модулем 2	**50**	**10**	**—**	**10**	**—**	**30**
Змістовий модуль 3. Успадкування і поліморфізм						
Тема 5. Успадкування. Ієрархія класів. Поняття похідного та базового класів. Захищені члени класу. Співвідношення атрибутів доступу в базовому і похідному класах. Ієрархія класів. Віртуальні базові класи. Контейнеризація. Конструктори і деструктори похідних класів. Перетворення вказівників на об'єкти класів. Приклади успадкування.	10	2	—	2	—	6
Тема 6. Поліморфізм. Віртуальні функції. Віртуальні функції, їх відмінність від звичайних функцій-членів. Віртуальні деструктори. Абстрактні функції і абстрактні класи. Поліморфні функції. Множинне успадкування. Створення бібліотеки класів графічних фігур.	10	2	—	2	—	6

Назви змістових модулів і тем	Усього годин (денна форма)	Лекції (л)	Практичні (п)	Лабораторні (лаб)	Інд. завд. (інд)	Самост. робота (с.р.)
1	**2**	**3**	**4**	**5**	**6**	**7**
Тема 7. Бібліотеки класів і шаблонів C++. Простори імен. Огляд можливостей бібліотек STL, BOOST, MTL. Робота з рядками засобами стандартної бібліотеки та без них. Узагальнене програмування і стандартна бібліотека шаблонів STL. Приклади використання бібліотеки STL.	10	2	—	2	—	6
Тема 8. Виняткові ситуації (Exception). Принципи обробки виключень. Створення та обробка власних класів-виключень. Стандартна ієрархія виключень (std::exception). Сучасні можливості C++ в обробці виключень (Modern C++).	10	2	—	2	—	6
Тема 9. Робота з потоками даних (Stream). Ієрархія класів потокового введення-виведення. Робота з файловими та рядковими потоками (<fstream>, <sstream>). Стандартні рядки C++ та їх маніпулювання через потоки. Ефективний парсинг та сучасне форматування (C++17 / C++20 / C++23). Практичний приклад: Парсинг та обробка потокових даних.	10	2	—	2	—	6
Разом за змістовим модулем 3	**50**	**10**	**—**	**10**	**—**	**30**
Усього годин за 3-й семестр	**150**	**30**	**—**	**30**	**—**	**90**

** ІНДЗ – для змістового модуля, або в цілому для навчальної дисципліни за рішенням кафедри (викладача).*

Тематика та питання лекційних занять

1. Тема 1. Парадигма програмування. Процедурне, модульне і об'єктно-орієнтоване програмування. Вступ. Парадигма програмування. Процедурне програмування. Модульне програмування. Об'єктно-орієнтоване програмування. Загальна характеристика мови C++ як мови, що найбільш втілює ідеї об'єктно-орієнтованого програмування.

2. Тема 2. Розширені можливості мови C++, що не пов'язані з класами. Прототипи функцій, аргументи за замовчуванням, посилання. Модифікатори const і volatile. Функціональний запис перетворення типів. Доступ до глобальних змінних, які закриті локальними. Функції, що підставляються. Оператори динамічного розподілу пам'яті. Перевантаження функцій, шаблони функцій. Перевантаження операторів для структур. Перевантаження операцій потокового вводу і виводу для структур. Приклади перевантаження операцій і функцій, використання шаблонів.

3. Тема 3. Класи і об'єкти. Створення і знищення об'єктів. Поняття класу та його визначення. Специфікатори public, private. Доступ до відкритих членів класу. Функції доступу до захованих членів класу. Створення об'єктів. Статичні члени класу. Дружні і складові функції. Дружні класи. Ініціалізація та знищення об'єктів. Автоматичне і динамічне виділення пам'яті під об'єкти класу. Конструктори і деструктори.

4. Тема 4. Операції над класовими об'єктами. Перевизначення операцій для класів. Бінарні і унарні операції. Пріоритети при перевизначенні операцій. Присвоєння і ініціалізації. Перевантаження операторів виклику функції, індексування. Стандартні об'єкти потокового вводу-виведення C++. Перевантаження операцій "<<", ">>" для класу користувача. Приклади проєктування і побудови класів.

5. Тема 5. Успадкування. Ієрархія класів. Поняття похідного та базового класів. Захищені члени класу. Співвідношення атрибутів доступу в базовому і похідному класах. Ієрархія класів. Віртуальні базові класи. Контейнеризація. Конструктори і деструктори похідних класів. Перетворення вказівників на об'єкти класів. Приклади успадкування.

6. Тема 6. Поліморфізм. Віртуальні функції. Віртуальні функції, їх відмінність від звичайних функцій-членів. Віртуальні деструктори. Абстрактні функції і абстрактні класи. Поліморфні функції. Множинне успадкування. Створення бібліотеки класів графічних фігур.

7. Тема 7. Бібліотеки класів і шаблонів C++. Простори імен. Огляд можливостей бібліотек STL, BOOST, MTL. Робота з рядками засобами стандартної бібліотеки та без них. Узагальнене програмування і стандартна бібліотека шаблонів STL. Приклади використання бібліотеки STL.

8. Тема 8. Виняткові ситуації (Exception). Принципи обробки виключень. Створення та обробка власних класів-виключень. Стандартна ієрархія виключень (std::exception). Сучасні можливості C++ в обробці виключень (Modern C++).

9. Тема 9. Робота з потоками даних (Stream). Ієрархія класів потокового введення-виведення. Робота з файловими та рядковими потоками (<fstream>, <sstream>). Стандартні рядки C++ та їх маніпулювання через потоки. Ефективний парсинг та сучасне форматування (C++17 / C++20 / C++23). Практичний приклад: Парсинг та обробка потокових даних.

Тематика та питання (завдання) лабораторних занять. Розподіл балів і годин

Модуль 1. Нові можливості C++, які не пов'язані з ООП (19 балів, 10 годин)

Елемент модуля	Кількість балів	Кількість годин
Лабораторна робота №1: Структури. Зовнішні функції і функції-члени структури	6 (3+3)	4 (2+2)
Лабораторна робота №2: Перезавантаження операцій і функцій для структур. Використання шаблонів	5	2
Лабораторна робота №3: Структури і класи	8 (4+4)	4 (2+2)

Модуль 2. Інкапсуляція. Поняття класу (28 балів, 10 годин)

Елемент модуля	Кількість балів	Кількість годин
Лабораторна робота №4: Створення класів. Конструктори, деструктори, дружні функції і члени класу. Перезавантаження операцій для класів	15 (5+5+5)	5
Індивідуальне практичне завдання: Створення умови задачі для лабораторної роботи №5 (успадкування або агрегація)	2	0,5
Лабораторна робота №5: Проєктування ієрархії класів. Успадкування. Агрегація	11	4

** Примітка: 0,5 год. аудиторного часу відводиться на написання контрольної задачі на тему «Створення класів», яка оцінюється у складі підсумкового контролю (10 балів).*

Модуль 3. Успадкування і поліморфізм (18 балів, 10 годин)

Елемент модуля	Кількість балів	Кількість годин
Лабораторна робота №6: Абстрактні класи	11	5
Лабораторна робота №7: Створення графічного інтерфейсу до лабораторної роботи №4, 5 або 6	7	5

Підсумковий контроль (залік): тестування Moodle – 25 балів (10 балів за тематикою першого модуля; 15 балів за тематикою другого і третього модулів); підсумкова контрольна задача на тему «Створення класів» – 10 балів.

Перелік питань для самоконтролю та контролю знань студентів

Тематика задач контрольної роботи

1. Робота з масивами структур (елементи структури – лише члени-дані, функції для роботи зі структурними змінними – зовнішні).

2. Створення класу та його використання. Клас включає члени-дані, конструктори, деструктор, функції-члени і функції-друзі класу, операції-члени і операції-друзі класу.

Теоретичні питання до підсумкового тестування

1. Нові можливості C++, які не пов’язані з ООП: доступ до глобальних змінних, які закриті локальними, модифікатор const, функціональний і операторний записи перетворення типів.

2. Нові можливості C++, які не пов’язані з ООП: прототипи функцій, аргументи за замовчуванням, посилання.

3. Нові можливості C++, які не пов’язані з ООП: функції, що підставляються, оператори динамічного розподілу пам’яті.

4. Нові можливості C++, які не пов’язані з ООП: перевантаження функцій, шаблони функцій.

5. Нові можливості C++, які не пов’язані з ООП: перевантаження операторів для структур (без <<, >>).

6. Нові можливості C++, які не пов’язані з ООП: перевантаження операцій потокового вводу і виводу для структур (“>>”, “<<”).

7. Інкапсуляція. Класи. Специфікатори public, private. Доступ до членів класу.

8. Статичні члени-дані, статичні члени-функції.

9. Дружні і складові (члени-функції) функції. Дружні класи.

10. Перевантаження операторів для класів. Бінарні і унарні операції. Дружні оператори.

11. Перевантаження операторів для класів: присвоєння і ініціалізація.

12. Перевантаження операторів для класів: індексування, операція виклику функції.

13. Ініціалізація і знищення, конструктори і деструктори.

14. Стандартні об’єкти потокового вводу-виведення C++. Перевантаження операцій “<<”, “>>” для класу користувача.

15. Автоматичне і динамічне виділення пам’яті під об’єкти класу. Функції доступу до захованих членів класу.

16. Шаблони класів.

17. Утворення нових класів з існуючих. Агрегація. Порядок виклику конструкторів і деструкторів при агрегації.

18. Успадкування. Похідні класи. Множинне успадкування.

19. Ієрархія класів. Специфікатори доступу. Співвідношення атрибутів доступу в базовому і похідному класах.

20. Успадкування: віртуальні базові класи.

21. Успадкування: перетворення вказівників на об’єкти класів.

22. Статичний та динамічний поліморфізм. Поліморфні класи.

23. Віртуальні функції-члени. Їх відмінність від звичайних функцій-членів.

24. Віртуальні деструктори. Поліморфні функції.

25. Абстрактні класи.

26. Огляд можливостей бібліотек STL, BOOST, MTL.

27. Робота з рядками засобами стандартної бібліотеки та без них.

28. Узагальнене програмування і стандартна бібліотека шаблонів STL. Приклади використання бібліотеки STL.

29. Поняття компонентного класу. Бібліотека VCL.

30. Побудова віконних інтерфейсів. Використання стандартних елементів керування.

31. Виняткові ситуації (Exception): принципи обробки виключень, створення та обробка власних класів-виключень, стандартна ієрархія виключень (std::exception), сучасні можливості C++ в обробці виключень.

32. Робота з потоками даних (Stream): ієрархія класів потокового введення-виведення, робота з файловими та рядковими потоками (<fstream>, <sstream>), ефективний парсинг та сучасне форматування.

Завдання для самостійної роботи

Самостійна робота студентів є важливою складовою навчального процесу і спрямована на поглиблення, узагальнення та закріплення знань, отриманих на лекційних і лабораторних заняттях, а також на формування навичок самостійного опрацювання теоретичного матеріалу, розроблення алгоритмів і програмних рішень.

До кожної теми навчальної дисципліни подаються види самостійної роботи, що враховують специфіку освітньої програми «Системний аналіз» і зміст навчальної дисципліни «Об'єктно-орієнтоване програмування».

Основні види самостійної роботи студентів:

- опрацювання конспектів лекцій, навчальної та наукової літератури з теми;
- виконання завдань, розміщених у системі дистанційного навчання (Moodle);
- підготовка до лабораторних і модульних робіт;
- розроблення індивідуальних класів, ієрархій класів та програмних модулів відповідно до завдань дисципліни;
- самостійне опрацювання прикладних можливостей мови C++ (графічні можливості, побудова віконних інтерфейсів) та їх застосування у лабораторних роботах;
- підготовка до тестового контролю та підсумкового заліку.

Кількість годин, передбачених навчальним планом для самостійної роботи, розподіляється між темами дисципліни відповідно до їх змісту та складності.

Орієнтовний розподіл самостійної роботи за темами:

№	Назва теми	Орієнтовна кількість годин самостійної роботи
1	Парадигма програмування. Процедурне, модульне і об'єктно-орієнтоване програмування	12
2	Розширені можливості мови C++, що не пов'язані з класами	18
3	Класи і об'єкти. Створення і знищення об'єктів	15
4	Операції над класовими об'єктами	15
5	Успадкування. Ієрархія класів	6
6	Поліморфізм. Віртуальні функції	6
7	Бібліотеки класів і шаблонів C++	6
8	Виняткові ситуації (Exception)	6
9	Робота з потоками даних (Stream)	6
	Разом:	**90 годин**

Теми для додаткового самостійного опрацювання:

№ п/п	Назва теми	Форма контролю
1	Графічні можливості	Застосування в лабораторних роботах
2	Створення віконних інтерфейсів (GUI)	Застосування в лабораторних роботах

Примітка: конкретний зміст і форми самостійної роботи уточнюються викладачем з урахуванням рівня підготовки студентів, особливостей навчального процесу та використання системи дистанційного навчання Moodle.

Критерії оцінювання результатів навчання з навчальної дисципліни

Підсумковий контроль (залік) проводиться засобами дистанційної системи Moodle.

Складові частини оцінки:

1. тести на 25 балів:

- 10 балів (тематика першого модуля);
- 15 балів (тематика другого і третього модулів).

2. програмно згенерована випадкова контрольна задача на тему «Створення класів» на 10 балів.

У сумі з лабораторними роботами (65 балів) це загалом складатиме 100 балів.

Захист лабораторної роботи та критерії оцінювання

Здача лабораторної роботи проводиться під час заняття згідно з календарним планом.

Для захисту лабораторної роботи кожен студент має самостійно виконати лабораторну роботу і здати її викладачу на занятті.

Не допускається заочне прийняття програм (електронною поштою) без запуску програм з різними вхідними даними.

Під час здачі програми викладач зобов'язаний перевіряти здатність студента орієнтуватися у власній програмі, пропонуючи йому виконати нескладні зміни, розраховані на 5–10 хвилин поточного заняття.

При необхідності виконання частини завдання або усього завдання у робочому зошиті, бали виставляти у зошиті, вказуючи число і підпис.

Під час здачі лабораторної роботи студент повинен:

- вміти пояснити постановку задач, які розв'язувались в лабораторній роботі; алгоритм розв'язування задач; програмну реалізацію завдання;
- продемонструвати розуміння програми та обґрунтувати зроблені висновки;
- відповісти на питання, які належать до виконання лабораторної роботи та додаткові теоретичні питання, якщо розданий перелік таких питань.

Якщо студент не розуміє алгоритму розв'язання задачі, не орієнтується в програмній реалізації, але є у наявності правильно виконувана програма, то робота зараховується не більше як на 30%.

Якщо студент розуміє задачу і алгоритм її виконання, але не орієнтується (слабо орієнтується) в практичній частині (програмній реалізації), то оцінка знижується до 50%.

Якщо програма не працює, або працює частково і студент може пояснити алгоритм, роботу оцінювати частково, залежно від об'єму і якості коду.

Кількість балів за лабораторну роботу визначає викладач у процесі здачі. Оцінка повідомляється студенту.

За невчасний захист лабораторних робіт у межах модуля допускається знімати по одному балу за кожне прострочене заняття, якщо робота оцінюється до 10 балів, і по 1,5–2 бали, якщо робота оцінюється в межах від 11 до 20 балів, але не більше половини балів.

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою (для заліку)
90–100	**A**	зараховано
80–89	**B**	зараховано
70–79	**C**	зараховано
60–69	**D**	зараховано
50–59	**E**	зараховано
35–49	**FX**	незараховано з можливістю повторного складання
0–34	**F**	незараховано з обов'язковим повторним вивченням дисципліни

Засоби оцінювання

Під час проведення лекцій використовуються пасивний та активний методи навчання. Консультаційна робота. Під час виконання студентами лабораторних робіт використовуються активні методи навчання. Проведення модульних контрольних робіт та навчальна робота під час прийому лабораторних робіт.

Форми поточного та підсумкового контролю

1. Контрольна робота.

2. Оцінювання на лабораторних заняттях в 3-х модулях.

3. Підсумкове тестування на заліку засобами системи Moodle.

Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота							Підсумковий тест + задача (залік)	Сума
Змістовий модуль 1			Змістовий модуль 2		Змістовий модуль 3			
T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**		
6	5	8	15	13	11	7	**35**	**100**

T1, T2 ... T9 – теми змістових модулів.

Рекомендована література

Базова (основна)

1. Сопронюк Т.М. Об'єктно-орієнтоване програмування на С++ : навч. посібник / Т. М. Сопронюк. – Чернівці : Чернівецький національний ун-т, 2013. – 175 с. (з грифом МОНУ, лист №1/11-17600 від 18.11.2013).

2. Sopronyuk T. Object-oriented programming in C++ : Textbook / Translated by Nonna Shulga. – CreateSpace, 2014. – 130 p. ISBN-13: 978-1502520906 (переклад навчального посібника з грифом

МОНУ: Сопронюк Т.М. Об’єктно-орієнтоване програмування на С++ : навч. посібник / Т. М. Сопронюк. – Чернівці : Чернівецький національний ун-т, 2014. – 176 с.).

3. 150 C++ Programming Assignments. Variants of tasks & Examples of the code : Textbook / Authored by Tatyana Sopronyuk, Translated by Nonna Shulga. – CreateSpace, 2015. – 73 p. ISBN-13: 978-1515254065, ISBN-10: 1515254062.

4. Sopronyuk T. C++ Programming: Theory and Assignments / Authored by Tatyana Sopronyuk, Translated by Nonna Shulga. – Independently published, 2020. – 178 p. ISBN-13: 979-8651045624.

5. Глинський Я.М., Анохін В.Є., Ряжська В.А. С++ і С++ Builder. – Львів : Деол, 2003. – 192 с.

6. Ткачук В.М. Програмування на С++: Лабораторний практикум. – Івано-Франківськ : Видавництво Прикарпатського національного університету імені Василя Стефаника, 2011. – 160 с.

7. Грицюк Ю.І., Рак Т.Є. Програмування мовою С++ : навчальний посібник. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 292 с.

8. Бєлов Ю.А., Карнаух Т.О., Коваль Ю.В., Ставровський А.Б. Вступ до програмування мовою С++. Організація обчислень : навч. посібник. – К. : Видавничо-поліграфічний центр «Київський університет», 2012. – 175 с.

9. Dewhurst, Stephen C. C++ Common Knowledge: Essential Intermediate Programming. – Upper Saddle River, N.J. : Addison-Wesley, 2005.

10. Schildt, Herbert. Herb Schildt's C++ Programming Cookbook. – New York : McGraw-Hill, 2008.

11. Savitch, Walter. Problem Solving with C++: The Object of Programming. – Fifth Edition (International ed.). – Boston : Pearson Addison Wesley, 2005.

12. Malik, D. S. C++ Programming: From Problem Analysis to Program Design. – Fourth Edition. – Boston, MA : Course Technology, 2009.

13. Savitch, Walter J. Problem Solving with C++. – Sixth Edition. – Boston : Pearson/Addison-Wesley, 2006.

Інформаційні ресурси

1. Відкритий освітній ресурс з програмування мовою С++:
   Ghunaim H. Introduction to C++. Volume I : open educational resource. Fort Hays State University, 2025. DOI: 10.58809/FNWV7666.

   https://fhsu.pressbooks.pub/IntroductionToCPlusPlusVolumeI/
2. Онлайн-посібник із сучасного С++ (Learn C++):
   https://www.learncpp.com/
3. Онлайн-ресурс із сучасних стандартів С++ (C++ Stories):
   https://www.cppstories.com/
4. Відкритий підручник з об’єктно-орієнтованого програмування на С++:
   Khufu: Object-Oriented Programming Using C++ : open textbook. University of Toronto, 2025.

https://www.eecg.utoronto.ca/~salma/projects/textbooks/

Додатково. Методичне забезпечення

- Сопронюк Т.М. Об’єктно-орієнтоване програмування на С++ : навч. посібник / Т. М. Сопронюк. – Чернівці : Чернівецький національний ун-т, 2013. – 175 с. (з грифом МОНУ, лист №1/11-17600 від 18.11.2013).
- Sopronyuk T. Object-oriented programming in C++ : Textbook / Translated by Nonna Shulga. – CreateSpace, 2014. – 130 p. ISBN-13: 978-1502520906 (переклад навчального посібника з грифом МОНУ: Сопронюк Т.М. Об’єктно-орієнтоване програмування на С++ : навч. посібник / Т. М. Сопронюк. – Чернівці : Чернівецький національний ун-т, 2014. – 176 с.) https://dspace.chnu.edu.ua/items/fe667704-8e73-4f3f-a69e-6b82dde9ac9c
- 150 C++ Programming Assignments. Variants of tasks & Examples of the code : Textbook / Authored by Tatyana Sopronyuk, Translated by Nonna Shulga. – CreateSpace, 2015. – 73 p. ISBN-13: 978-1515254065, ISBN-10: 1515254062. https://dspace.chnu.edu.ua/items/c8ff191f-5219-4311-bafe-d70298c21862
- Sopronyuk T. C++ Programming: Theory and Assignments / Authored by Tatyana Sopronyuk, Translated by Nonna Shulga. – Independently published, 2020. – 178 p. ISBN-13: 979-8651045624. https://www.amazon.sg/Programming-Theory-Assignments-Tetyana-Sopronyuk/dp/B089M41MRY
- Тестові завдання (Система Moodle).
- Презентації лекцій (Система Moodle).
- Відео-лекції на Google диску.

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- «Кодекс академічної доброчесності та етики Чернівецького національного університету імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/kodeks-akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yuriia-fedkovycha/
- «Положення про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyiavlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/
- «Політика використання штучного інтелекту в Чернівецькому національному університету імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/

Використання здобувачами освіти інструментів штучного інтелекту (ШІ) під час виконання навчальних завдань:

- допускається за умови дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов’язаний зазначити факт і спосіб його застосування);
- не допускається використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- додаткові обмеження чи вимоги щодо використання ШІ можуть визначатися викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.