

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
В. о. декана факультету комп'ютерних
інформаційних технологій
Ігор ЯКИМЕНКО
"_____ 2023 р.

ЗАТВЕРДЖУЮ
В. о. проректора з науково-
педагогічної роботи
Віктор ОСТРОВЕРХОВ
"_____ 2023 р.

ЗАТВЕРДЖУЮ
Директор навчально-наукового
інституту новітніх освітніх технологій
Святослав ПИТЕЛЬ
"_____ 2023 р.

РОБОЧА ПРОГРАМА

з дисципліни «**Прикладне програмне забезпечення для комп'ютерних систем та мереж**»

ступінь вищої освіти – бакалавр

галузь знань – 12 “Інформаційні технології”

спеціальність – 123 “Комп'ютерна інженерія”

освітньо-професійна програма – „Комп'ютерна інженерія”

Кафедра комп'ютерної інженерії

Форма навчання	Курс	Семестр	Лекції (год.)	Лабораторні (год.)	ІРС (год.)	Тренінг, КПЗ (год.)	Самост. робота студ. (год.)	Разом (год.)	Залік (сем.)	Екз. (сем.)
Денна	1	2	46	30	5	10	59	150	-	2
Заочна	1	2	8	4	0	0	138	150	-	2

31.08.2023
[Підпис]

Тернопіль – ЗУНУ
2023

Робоча програма складена на основі освітньо – професійної програми підготовки бакалавра галузі знань 12 “Інформаційні технології” спеціальності 123 “Комп’ютерна інженерія”, затвердженої Вченою радою ЗУНУ (протокол № 10 від 23 червня 2023 р.).

Робочу програму склав к.т.н., доцент кафедри КІ

Юрій БАТЬКО

Робоча програма затверджена на засіданні кафедри комп’ютерної інженерії, протокол №1 від 28 серпня 2023р.

Завідувач кафедри



Леся ДУБЧАК

Розглянуто та схвалено групою забезпечення спеціальності «Комп’ютерна інженерія», протокол №1 від 28 серпня 2023 р.

Голова ГЗС



Олег БЕРЕЗЬКИЙ

Гарант ОП



Леся ДУБЧАК

**СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
" ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ КОМП'ЮТЕРНИХ
СИСТЕМ ТА МЕРЕЖ"**

1. Опис дисципліни " Прикладне програмне забезпечення для комп'ютерних систем та мереж "

Дисципліна – «Прикладне програмне забезпечення для комп'ютерних систем та мереж»	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів – <i>Денна – 5,</i> <i>Заочна – 5</i>	галузь знань – 12 „Інформаційні технології”	Статус дисципліни – нормативна Мова навчання – українська
Кількість залікових модулів: 4	Спеціальність – 123 „Комп'ютерна інженерія”	Рік підготовки: <i>Денна - 1,</i> <i>Заочна – 1</i> Семестр: <i>Денна – 2</i> <i>Заочна – 2</i>
Кількість змістових модулів – 3	Ступінь вищої освіти - бакалавр	Лекції: <i>Денна - 46 год.,</i> <i>Заочна – 8 год.</i> Лабораторні заняття: <i>Денна - 30 год.</i> <i>Заочна – 4 год.</i>
Загальна кількість годин – <i>Денна – 150 год.,</i> <i>Заочна - 150 год.</i>		Самостійна робота: <i>Денна – 59 год.</i> <i>Заочна – 138 год.</i> Тренінг, КПІЗ – 10 год. Індивідуальна робота: <i>Денна - 5 год.</i>
Тижневих годин: <i>Денна: 2 семестр:</i> 10 год., з них аудиторних – 5 год.		Вид підсумкового контролю <i>Денна: 2 семестр – екзамен</i> <i>Заочна: 2 семестр – екзамен</i>

2 Мета й завдання дисципліни

" Прикладне програмне забезпечення для комп'ютерних систем та мереж "

2.1 Мета вивчення дисципліни

Програма та тематичний план дисципліни „Прикладне програмне забезпечення для комп'ютерних систем та мереж” орієнтовані на глибоке та ґрунтовне засвоєння студентами систематичних знань та практичних навичок для створення програмних продуктів з ефективним використанням сучасних технологій.

Метою дисципліни є вивчення науково-практичного інструментарію технологій структурного та об'єктно-орієнтованого підходів при проектуванні комп'ютерних програм та його використанні при реалізації програмних додатків.

2.2 Завдання вивчення дисципліни

Завданням вивчення дисципліни є вивчення науково-практичного інструментарію проектування прикладних програмних додатків та їх складових за допомогою сучасних засобів автоматизованого проектування.

В результаті вивчення дисципліни студенти мають вміти визначати алгоритми вирішення поставлених задач, проектувати загальну структуру програмних додатків, реалізовувати поставлені задачі за допомогою основних бібліотек та команд мови C++.

2.3 Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни:

K12. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

K13. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

2.4 Передумови для вивчення дисципліни

Зазначена дисципліна включена до циклу дисциплін професійної підготовки за переліком програми. У структурно-логічній схемі навчання зазначена дисципліна розміщена на I-му курсі. Вивчення курсу „Прикладне програмне забезпечення для комп'ютерних систем та мереж” передбачає наявність систематичних та ґрунтовних знань із суміжних курсів («Основи програмування»), а також цілеспрямованої роботи на лекційних та лабораторних заняттях, самостійної роботи студентів.

2.5 Результати навчання

В результаті вивчення дисципліни студенти повинні:

ПРН1. Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.

ПРН2. Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах.

ПРН10. Вміти розробляти програмне забезпечення для вбудованих і

розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

ПРН16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

3. Зміст дисципліни

«Прикладне програмне забезпечення для комп'ютерних систем та мереж»

Змістовий модуль 1. Об'єктно-орієнтоване програмування.

Тема 1. Об'єктно-орієнтоване програмування.

1. Об'єктно-орієнтовані технології. 2. Парадигми ООП: інкапсуляція, поліморфізм, наслідування. 3. Ієрархія.

Література: 4, 7.

Тема 2. Поняття класу.

1. Поняття класу. 2. Поняття екземпляру класу. 3. Конструктор та Деструктор класу. Конструктор копіювання. 4. Поля та методи класів.

Література: 7, 8.

Тема 3. Поняття інкапсуляції.

1. Поняття інкапсуляції. Переваги та недоліки використання інкапсуляції. 2. Модифікатори доступу. 3. Використання інкапсуляції при проектуванні класів. Сеттери та геттери при реалізації класу.

Література: 2, 3, 7, 8.

Тема 4. Принцип наслідування в об'єктно-орієнтовано підході.

1. Ієрархія класів. 2. Базовий клас. 3. Абстрактний клас. 4. Модифікатори доступу при наслідуванні. 5. Переваги та недоліки використанні наслідування.

Література: 2, 3, 7, 8.

Тема 5. Поліморфізм.

1. Поняття поліморфізму. 2. Пізніє та ранніє зв'язування. 3. Віртуальний метод. 4. Чисто віртуальні методи. 5. Абстрактні класи. 6. Дружні функції. 7. Дружні методи. 8. Таблиця віртуальних методів.

Література: 2, 8.

Тема 6. Перезавантаження унарних операторів.

1. Принципи перезавантаження операторів. 2. Перезавантаження оператора присвоєння (=). 3. Перезавантаження оператора інкременту (префіксна та постфіксна форми), 4. Перезавантаження оператора декретенту (префіксна та постфіксна форми).

Література: 4, 8.

Тема 7. Перезавантаження бінарних операторів.

1. Перезавантаження математичних операторів (+, -, *, /, %). 2. Перезавантаження математичних операторів з присвоєнням (+=, -=, *=, /=, %=). 3. Перезавантаження

логічних операторів (>, <, !=, ==, >=, <=). 4. Перезавантаження операторів вводу/виводу (<</>>). 5. Перезавантаження операторів індексації ([], ()).

Література: 4, 5, 7.

Тема 8. Шаблони класів.

1. Поняття шаблону класу. 2 Контейнерні класи. 3. Шаблони класів в стандартній бібліотеці класів. 4. Шаблони класів та заголовочні файли.

Література: 4, 5, 10.

Змістовий модуль 2. Алгоритми і структури даних.

Тема 9. Standard Template Library принципи, вимоги, типи

1. Поняття колекції (vector, list, map, set, multiset, multimap) 2.STL Ітератори. 3. Поняття ітераторів. 4. Типи ітераторів. 5. Вимоги та реалізація.

Тема 10. Алгоритми бібліотеки STL

1. Методи перебору та обробки колекцій. 2. Методи сортування елементів колекцій. 3. Методи виконання математичних операцій над елементами колекцій.

Література: 7, 9.

Тема 11. Задачі обробки елементів масивів.

1. Задачі обробки масивів. 2. Алгоритми сортування масивів. 3 Алгоритм сортування вибором. 4. Алгоритм сортування вставкою. 5. Алгоритм сортування бульбашкою. 6. Алгоритм швидкого сортування. 7. Методи qsort() та sort().

Література: 4, 7, 8.

Тема 12. Задачі пошуку елементів в масивах даних.

Поняття пошуку елемента в масиві. 2. Класифікація алгоритмів пошуку. 3. Лінійний пошук. 4. Алгоритм бінарного пошуку. 5. Алгоритм пошуку стрибками. 6. Алгоритми пошуку на основі індексації.

Література: 7, 9.

Тема 13. Рекурсія.

1. Поняття рекурсії. 2. Принципи організації рекурсії. 3. Переваги та недоліки використання рекурсії. 4. Форми рекурсії. 5. Лінійна рекурсія. 6. Повторна рекурсія. 7. Каскадна рекурсія. 8. Віддалена рекурсія. 9. Взаємна рекурсія. 10. Рекурсивні та ітеративні описи алгоритмів.

Література: 7, 9.

Тема 14. Лямбда вирази.

1. Введення в лямбда-вирази. 2.Тип лямбда-виразів. 3.Загальні/Узагальнені лямбди. 4. Загальні лямбди та статичні змінні. 5. Виведення типу, що повертається, і типи, що повертаються trailing.

Література: 4, 6, 9.

Тема 15. Списки.

1. Поняття списку в програмуванні. 2. Принципи організації списків. 3. Переваги та недоліки використання списків. 4. Види списків.

Література: 7, 9.

Тема 16. Дерева.

Поняття дерева. Двійкове дерево. Дерево пошуку. Обхід двійкового дерева. Обчислення арифметичних виразів. Використання пов'язаних структур. Зберігання двійкового дерева в масиві.

Література: 3, 7, 9.

Тема 17. Тип даних стек.

1. Сегменти в програмі. 2. Куча. 3. Стек викликів. 4. Стек як структура даних. 5. Сегмент стеку викликів.

Література: 4, 6, 9.

Змістовий модуль 3. Стандартні та нестандартні виключення.

Тема 18. Обробка виключень.

1. Поняття виняткової ситуації. 2. Оператор try. 3. Оператор Throw. 4. Оператор catch.

Література: 2, 4.

Тема 19. Робота з файловою системою операційної системи.

1. Поняття файлової системи. 2. Пошук файлів. 3. Обробка файлів. 4. Робота з каталогами.

Література: 2, 4.

Тема 20. Бібліотека роботи з графічними елементами SFML.

1. Поняття технології SFML. 2. Принципи підключення функцій бібліотеки SFML в проект. 3. Створення візуального вікна для відображення графічних елементів. 4. Відображення графічних примітивів за допомогою функцій бібліотеки SFML.

Література: 4, 8.

Тема 21. Проектування динамічних сцен на основі бібліотеки SFML.

1. Принципи створення динамічної сцени. 2. Поняття текстури. 3. Поняття спрайта. 4. Поняття колізії. 5. Додавання аудіо складової.

Література: 4, 8.

**забезпечення для комп'ютерних систем та мереж»
(денна форма навчання)**

	Кількість годин					Контрольні заходи
	Лекції	Лабораторні заняття	Самостійна робота	Індивід. робота	Трейнінг, КПЗ	
Змістовий модуль 1						
Тема 1. Об'єктно-орієнтоване програмування	2		4	1	4	опитування
Тема 2. Поняття класу	2	6	4	1		опитування
Тема 3. Поняття інкапсуляції	2		4			опитування
Тема 4. Принцип наслідування в об'єктно-орієнтовано підході	2	6	6			опитування
Тема 5. Поліморфізм	2		4			опитування
Тема 6. Перезавантаження унарних операторів.	2	6	4			опитування
Тема 7. Перезавантаження бінарних операторів.	2		2			опитування
Тема 8. Шаблони класів.	2		2			опитування
Змістовий модуль 2						
Тема 9. Standard Template Library принципи, вимоги, типи	2	6	4		4	опитування
Тема 10. Алгоритми бібліотеки STL	2		2	1		опитування
Тема 11. Задачі обробки елементів масивів.	2	6	4			опитування
Тема 12. Задачі пошуку елементів в масивах даних.	2		2	1		опитування
Тема 13. Рекурсія.	2		2			опитування
Тема 14. Лямбда вирази.	2		2			опитування
Тема 15. Списки	2		2			опитування
Тема 16. Дерева	2		2			опитування
Тема 17. Тип даних стек.	2		2			опитування
Змістовий модуль 3						
Тема 18. Обробка виключень	2		2	1	2	опитування
Тема 19. Робота з файловою системою операційної системи	2		2			опитування
Тема 20. Бібліотека роботи з графічними елементами SFML.	4	6	2			опитування
Тема 21. Проектування динамічних сцен на основі бібліотеки SFML.	4		1			опитування
Разом	46	30	59	5	10	

(заочна форма навчання)

	Кількість годин				
	Лекції	Лабораторні заняття	ІРС	Тренінг, КПЗ	Самостійна робота
Змістовий модуль 1					
Тема 1. Об'єктно-орієнтоване програмування	2				6
Тема 2. Поняття класу		2			8
Тема 3. Поняття інкапсуляції	2				6
Тема 4. Принцип наслідування в об'єктно-орієнтовано підході					8
Тема 5. Поліморфізм					8
Тема 6. Перезавантаження унарних операторів.					8
Тема 7. Перезавантаження бінарних операторів.					6
Тема 8. Шаблони класів.					6
Змістовий модуль 2					
Тема 9. Standard Template Library принципи, вимоги, типи	2	2			8
Тема 10. Алгоритми бібліотеки STL					6
Тема 11. Задачі обробки елементів масивів.					6
Тема 12. Задачі пошуку елементів в масивах даних.					6
Тема 13. Рекурсія.					6
Тема 14. Лямбда вирази.					6
Тема 15. Списки					6
Тема 16. Деревя					6
Тема 17. Тип даних стек.					6
Змістовий модуль 3					
Тема 18. Обробка виключень					6
Тема 19. Робота з файловою системою операційної системи	2				8
Тема 20. Бібліотека роботи з графічними елементами SFML.					6
Тема 21. Проектування динамічних сцен на основі бібліотеки SFML.					6
Разом	8	4			138

5. Тематика лабораторних занять

Лабораторна робота №1.

Тема: Парадигми ООП та проектування класів.

Мета: Отримати практичні навички по створенні та роботі з класами на мові C++.

Питання для обговорення:

1. Парадигми ООП.

2. Поняття класу.
3. Модифікатори доступу до полів та методів класу.

Література: 2, 4.

Лабораторна робота №2.

Тема: Проектування ієрархії класів на мові програмування C++.

Мета: Отримати практичні навички по проектуванні ієрархії класів на мові програмування C++.

Питання для обговорення:

1. Поняття наслідування
2. Одинарне та множинне наслідування.
3. Модифікатори доступу при наслідуванні.

Література: 9, 11, 12.

Лабораторна робота №3.

Тема: Перезавантаження операторів на мові програмування C++.

Мета: Отримати практичні навички по перезавантаженню операторів на мові програмування C++

Питання для обговорення:

1. Поняття перезавантаження операторів
2. Перезавантаження унарних операторів.
3. Перезавантаження бінарних операторів.

Література: 8, 10, 16.

Лабораторна робота №4.

Тема: Алгоритми сортування та пошуку

Мета: Вивчення сучасних алгоритмів пошуку та сортування засобами мови C++

Питання для обговорення:

1. Основні операції над масивами
2. Алгоритми сортування.
3. Алгоритми пошуку.

Література: 7, 8.

Лабораторна робота №5.

Тема: Робота з бібліотекою STL

Мета: Вивчення базових функцій бібліотеки STL

Питання для обговорення:

1. Основні колекції STL
2. Обробка векторів.
3. Алгоритми STL.

Література: 10, 11.

Лабораторна робота №6.

Тема: Графічна бібліотека SFML та C++

Мета: Створення програмних додатків з графічними елементами з використанням бібліотеки SFML

Питання для обговорення:

1. Основні структура даних SFML
2. Робота з геометричними примітивами.
3. Принципи створення анімації.

Література: 9, 15.

6. Комплексне практичне індивідуальне завдання

Комплексні практичні індивідуальні завдання (КПЗ) виконуються самостійно кожним студентом згідно з варіантом, отриманим у викладача. КПЗ з дисципліни «Прикладне програмне забезпечення для комп'ютерних систем та мереж» включає в себе такі завдання:

1. Теоретичний блок.
2. Розробка блок-схеми алгоритму.
3. Розробка програми для:
4. Розв'язання математичного виразу.
5. Розробка програми для обробки масиву даних.
6. Розробка програми для обробки стрічкової інформації.
7. Розробка програми для роботи з файлами.
8. Розробка програми на основі об'єктно-орієнтованого підходу.
9. Розробка програми зі складною ієрархією класів.
10. Розробка програми для роботи з графічними даними.

7. Самостійна робота студентів

(денна форма навчання)

№ п/п	Тематика
1	Змістовне поняття алгоритму та його основні властивості.
2	Абстракція даних.
3	Оператори управління МП.
4	Оператори вибору: типи, синтаксис і семантика. Техніка програмування розгалужень.
5	Оператори повторення: типи, синтаксис і семантика. Техніка програмування циклів.
6	Передумови, постумови і інваріанти циклів.
7	Методологія процедурного програмування.
8	Форми рекурсії. Лінійна рекурсія. Повторна рекурсія. Каскадна рекурсія.
9	Рекурсивні та ітеративні описи алгоритмів.
10	Одномірні масиви. Багатомірні масиви і загальні типи індексів. Динамічні і гнучкі масиви.
11	Задачі обробки масивів. Лінійний пошук у масиві. Бінарний пошук у масиві. Алгоритми сортування масивів.
12	Прості алгоритми сортування: сортування обмінами (бульбашкове), сортування

	вибором.
13	Робота з файлами. Текстові та двійкові файли.
14	Прямий доступ. Посилальний тип даних. Операції над вказівниками.
15	Розподіл пам'яті.
16	Динамічні інформаційні структури та їх реалізація.
17	Послідовності як абстрактні типи даних.
18	Обчислювальні структури послідовностей. Списки. Лінійні списки.
19	Структура програми на C++ та етапи її обробки.
20	Поняття компілятора, інтерпретатора.

(заочна форма навчання)

№ п/п	Тематика
1	Змістовне поняття алгоритму та його основні властивості.
2	Абстракція даних.
3	Оператори управління МП.
4	Оператори вибору: типи, синтаксис і семантика. Техніка програмування розгалужень.
5	Оператори повторення: типи, синтаксис і семантика. Техніка програмування циклів.
6	Передумови, постумови і інваріанти циклів.
7	Методологія процедурного програмування.
8	Форми рекурсії. Лінійна рекурсія. Повторна рекурсія. Каскадна рекурсія.
9	Рекурсивні та ітеративні описи алгоритмів.
10	Одномірні масиви. Багатомірні масиви і загальні типи індексів. Динамічні і гнучкі масиви.
11	Задачі обробки масивів. Лінійний пошук у масиві. Бінарний пошук у масиві. Алгоритми сортування масивів.
12	Прості алгоритми сортування: сортування обмінами (бульбашкове), сортування вибором.
13	Робота з файлами. Текстові та двійкові файли.
14	Прямий доступ. Посилальний тип даних. Операції над вказівниками.
15	Розподіл пам'яті.
16	Динамічні інформаційні структури та їх реалізація.
17	Послідовності як абстрактні типи даних.
18	Обчислювальні структури послідовностей. Списки. Лінійні списки.
19	Структура програми на C++ та етапи її обробки.
20	Поняття компілятора, інтерпретатора.
21	Заголовні файли.
22	Правила «гарного стилю» в програмуванні.
23	Система числення. Подання чисел в різних системах числення. Двійкова, шістнадцяткова система.
24	Константи, змінні, типи даних. Класифікація типів. Перетворення типів.
25	Машинне подання дійсних типів.

8 Організація і проведення тренінгу з дисципліни «Прикладне програмне забезпечення для комп'ютерних систем та мереж»

№п/п	Вид роботи	Порядок проведення тренінгу
1	Огляд сучасних середовищ розробки комп'ютерних програм	1. розгляд сучасних середовищ розробки фірм Bourland, Microsoft; 2. дослідження бібліотек програмних функції для створення програмних додатків по роботі з числовими та символьними змінними (math, string, iostream, sfml)
2	Розгляд процесу розробки алгоритму роботи програмних додатків на основі блок-схем	<ul style="list-style-type: none">• постановка задачі;• опис технічного завдання;• графічне відображення алгоритму роботи програми
3	Реалізація розробленого алгоритму роботи програми на мові програмування С++	1. Реалізація алгоритму роботи програми на мові програмування С++; 2. реалізація інтерфейсу користувача.
4	Тестування розробленого програмного додатку	<ul style="list-style-type: none">– вибір та обґрунтування тестової вибірки;– опис вхідних та вихідних даних;– перевірка правильності роботи реалізованого програмного додатку

9 Засоби оцінювання та методи демонстрування результатів навчання

У навчальному процесі застосовуються: лекції, в тому числі з використанням мультимедіапроектора та інших ТЗН; практичні заняття; індивідуальні заняття, самостійна робота студента, робота в Інтернет.

У процесі вивчення дисципліни "Технології програмування для комп'ютерних систем та мереж" використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточні опитування;
- залікове модульне тестування та опитування;
- презентації результатів виконання завдань та досліджень;
- оцінювання результатів КПЗ;
- завдання на лабораторному обладнанні;
- ректорська контрольна робота;
- екзамен.

10. Критерії, форми поточного та підсумкового контролю

В процесі вивчення дисципліни «Прикладне програмне забезпечення для комп'ютерних систем та мереж» використовуються наступні методи оцінювання навчальної роботи студента:

- поточне тестування та опитування;
- підсумкове тестування по кожному змістовому модулю;
- ректорська контрольна робота;
- підсумкова оцінка за комплексне практичне індивідуальне завдання (КПЗ), враховуючи поточне опитування;
- підсумковий письмовий екзамен.

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Прикладне

програмне забезпечення для комп'ютерних систем та мереж» визначається як середньозважена величина, в залежності від питомої ваги кожної складової залікового кредиту:

2 семестр

Заліковий модуль 1	Заліковий модуль 2	Заліковий модуль 3 (підсумкова оцінка за КПІЗ)	Заліковий модуль 4 (екзамен)	Разом
20 %	20 %	20 %	40 %	100%
1. Усне опитування під час заняття (10 тем по 5 бали = 50 балів) 2. Письмова робота = 50 балів	1. Усне опитування під час заняття (11 тем по 5 балів = 55 балів) 2. Письмова робота = 45 балів	1. Написання та захист КПІЗ = 80 балів. 2. Виконання завдань під час тренінгу = 20 балів	1. Тестові завдання (25 тестів по 2 бали за тест) – макс. 50 балів 2. Завдання. 1 – макс. 25 балів 3. Завдання. 2 – макс. 25 балів	100

Шкала оцінювання

За шкалою університету	За національною шкалою	За шкалою ECTS
90-100	Відмінно	A (відмінно)
85-89	Добре	B (дуже добре)
75-84		C (добре)
65-74	Задовільно	D (задовільно)
60-64		E (достатньо)
35-59	Незадовільно	FX (незадовільно, з можливістю повторного складання)
1-34		F (незадовільно, з обов'язковим повторним курсом)

11. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1.	DEV C++	1-21
2.	MS Visual Studio Community Edition	1-21

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Alexander Stepanov and Paul McJones Elements of Programming Semigroup Press, 2019, 218
2. Kenneth Leroy Busbee, Dave Braunschweig. Programming Fundamentals - A Modular Structured Approach, 2nd Edition Houston Community College; eBook (Creative Commons Licensed)
3. Changkun Ou Modern C++ Tutorial: C++11/14/17/20 On the Fly - The Fastest Guide towards Modern C++ GitHub; eBook (Creative Commons Licensed), 92.
4. Johan Sannemo Principles of Algorithmic Problem Solving KTH Royal Institute of Technology, 2018, 351.
5. Divakar Viswanath Scientific Programming and Computer Architecture. The MIT Press, 2017, 434.
6. Frank B. Brokken C++ Annotations: An Extensive Tutorial about the C++ Programming Language. University of Groningen, 2021.
7. Bjarne Stroustrup, Herb Sutter C++ Core Guidelines Standard C++ Foundation and GitHub. 2022.
8. C++ Notes for Professionals. Stack Overflow Contributors. Goalkicker. 2018, 708.
9. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi How to Design Programs, second edition: An Introduction to Programming and Computing (The MIT Press) second edition 2018, 792.
10. Roldan Pozo, Ronald Kriemann, et al. C++ Programming for Scientists NIST, Ohio Supercomputer Center
11. Dr. Rian Quinn. Hands-On System Programming with C++: Build Performant and Concurrent Unix and Linux Systems with C++17. Packt Publishing 2018, 522.
12. Florian Jatton The Constitution of Algorithms: Ground-Truthing, Programming, Formulating. The MIT Press, 2021, 154.
13. Xinyu Liu. Elementary Algorithms. GitHub.com, 2021, 503.
14. Granville Barnett, Luca Del Tongo, John Bullinaria. Data Structures and Algorithms: Annotated Reference with Examples. University of Birmingham, 2019, 112.