Чернівецький національний університет імені Юрія Федьковича

(повне найменування головного закладу вищої освіти)

Відокремлений структурний підрозділ «Фаховий коледж

Чернівецького національного університету імені Юрія Федьковича»

(повна назва відокремленого структурного підрозділу)

Циклова комісія *комп'ютерної інженерії*

(назва циклової/предметної комісії)

"ПОГОДЖЕНО"
Завідувач
Природничого відділення
Христина МЕЛЬНИЧУК
(підпис) (ім'я та прізвище)
"27" 08 20 26 року

"ЗАТВЕРДЖУЮ"
Заступник директора
з навчально-методичної роботи
Микола ДЕРЕВ'ЯНЧУК
(підпис) (ім'я та прізвище)
"29" 08 20 26 року

ПРОГРАМА

навчальної дисципліни

"Програмування Ч.2 Об'єктно-орієнтоване програмування"

(вкажіть назву навчальної дисципліни (іноземною, якщо дисципліна викладається іноземною мовою))

обов'язкова

(вказати: обов'язкова/вибіркова)

Освітньо-професійна програма

"Комп'ютерна інженерія"

(назва освітньо-професійної програми)

Спеціальність

F7 Комп'ютерна інженерія

(код і назва спеціальності)

Галузь знань

F Інформаційні технології

(код і назва галузі знань)

Освітньо-професійний ступінь

фаховий молодший бакалавр

(назва освітньо-професійного ступеня)

Мова навчання

українська

(вказати: на якій мові читається предмет)

Чернівці, 2026

Програма навчальної дисципліни *"Програмування Ч.2 Об'єктно-орієнтоване програмування"*

(назва навчальної дисципліни)

складена відповідно до освітньо-професійної програми *"Комп'ютерна інженерія"*

(назва освітньо-професійної програми)

затвердженої Вченою радою Чернівецького національного університету імені Юрія Федьковича (Протокол № 13 від «28» серпня 2026 року) та введеної в дію наказом ректора №315 від «28» серпня 2026 року.

Розробники: (вказати авторів, їхні посади, наукові ступені та вчені (педагогічні) звання)

викладач вищої категорії, к. фіз.-мат. н., доцент, В.В. Лазорик

Профайл викладача https://cs-college.chnu.edu.ua/tsyklova-komisiia/sklad-tsyklovoi-komisii/lazoryk-vasyl-vasylovych-vykladach-sumisnyk/

E-mail v.lazoryk@chnu.edu.ua

Сторінка курсу в Moodle https://moodle.chnu.edu.ua/course/view.php?id=552

Робоча програма обговорена та узгоджена на засіданні циклової комісії *комп'ютерної інженерії*

Протокол № 1 від "27" 08 2026 року

Голова циклової комісії _____ *Олександр ТАЩУК*

(підпис) (ім'я та прізвище)

Схвалена Методичною радою

Протокол № 1 від "27" 08 2026 року

Голова методичної ради _____ *Ольга БІЛОКРИЛА*

(підпис) (ім'я та прізвище)

ПЕРЕЗАТВЕРДЖЕНО

Протокол № ____ від _____, 20 __ р. _____ _____

(підпис) (ініціали та прізвище голови ЦК)

Протокол № ____ від _____, 20 __ р. _____ _____

(підпис) (ініціали та прізвище голови ЦК)

Протокол № ____ від _____, 20 __ р. _____ _____

(підпис) (ініціали та прізвище голови ЦК)

1. Мета та завдання навчальної дисципліни

Анотація: За останні кілька років об’єктно-орієнтована технологія проникла в різні розділи інформатики та комп’ютерної техніки. Сьогодні об’єктно-орієнтована технологія стала однією з основних засобів при розробці програмного забезпечення промислового масштабу. Ми бачимо, що в усьому світі об'єктна парадигма застосовується в таких різних областях, як керування банківськими трансакціями, автоматизація кегельбанів, керування комунальним господарством і дослідження генів людини. У багатьох випадках нові покоління операційних систем, систем керування базами даних, телефонних служб і мультимедіа - програм пишуться в об’єктно-орієнтованому стилі. У більшості таких випадків розробники проектів бажають використовувати об'єктно-орієнтовану технологію просто тому, що не було іншої можливості створити досить надійну і життєздатну систему. Об’єктно- орієнтоване проектування — це методика, що концентрує основну увагу програміста на зв'язках між об'єктами, а не на деталях їхньої реалізації. Вивчати програмування поза конкретною мовою недоцільно. З погляду нові технології та зв'язок з іншими мовами програмування, краще зупинити огляд на C++. Крім того, це - відкритий стандарт для мережних обчислень, порівняно проста для вивчення мова і мова навчання об’єктно-орієнтованому програмуванню.

Матеріал дисципліні сприяє набуттю вмінь розробки програм. За допомогою одержаних знань, вмінь та навичок студенти зможуть розробляти програмні продукти з використанням об’єктно-орієнтованих технологій програмування, стати досвідченими спеціалістами з розробки потужних програм та знавцями сучасних інформаційних технологій. Навчитися розробляти програмні складні програмні продукти використовуючи об’єктно-орієнтований підхід до програмування використовуючи стандартну бібліотеку шаблонів мови C++.

Мета дисципліни – поглиблене вивчення об’єктно-орієнтованого програмування та оволодіння навиками об’єктної технології для розв’язання складних задач. При розв’язанні складних задач від студента (майбутнього програміста) потрібно вміти представляти потоки інформації, аналізувати сучасні обчислювальні інструменти. У даній дисципліні студенти навчаться проводити об'єктно- орієнтований аналіз предметної області; здійснювати об’єктно-орієнтоване проектування; розробляти програмні продукти з використанням об’єктно-орієнтованої мови програмування C++. Крім того, студенти будуть розвивати свою пам'ять, логічне і просторове мислення, уяву та робити акценти на практичному застосуванні знань із об’єктно-орієнтованого програмування в загальному і C++ зокрема. Процес поглибленого вивчення об’єктно-орієнтованого програмування супроводжується детальним освоєнням стандартних бібліотек мови C++.

Основними завданнями

За допомогою одержаних знань, вмінь та навичок студенти зможуть розробляти програмні продукти з використанням об’єктно - орієнтованих технологій програмування, стати досвідченими спеціалістами з розробки потужних програм та знавцями сучасних інформаційних технологій. Навчитися розробляти програмні складні програмні продукти використовуючи об’єктно – орієнтований підхід до програмування використовуючи стандартну бібліотеку шаблонів мови C++.

В результаті вивчення дисципліни фахівець повинен знати:

– об’єктно - орієнтований підхід до програмування;

– основи мови програмування C та C++;

– оператори та операції мови;

– змінні, масиви, вказівники та посилання;

– функції;

– структур;

– директиви препроцесора;

– файли коду програми, файли заголовку, проектні файли;

– простори імен;

– класи, дані та методи класів;

– конструктори та деструктори класів;

– перевантаження операцій мови С++;

– успадкування;

– механізми віртуалізації, пізнє динамічне з'єднання;

– шаблони (параметризовані типи);

– обробка виключних ситуацій;

– стандартну бібліотеку шаблонів.

В результаті вивчення дисципліни фахівець повинен вміти:

– створювати та налагоджувати програми в середовищі Microsoft Visual Studio.

– розробляти алгоритми та складати програми обчислення математичних виразів;

– розробляти алгоритми та складати програми обчислення рекурентних виразів;

– розробляти алгоритми та складати програми обробки символьної інформації;

– розробляти алгоритми та складати програми обробки векторів та масивів даних;

– розробляти алгоритми та складати програми обробки даних із побітових операцій;

– розробляти алгоритми та складати програми для структурованих типів даних;

– розробляти алгоритми та складати програми обробки даних із використанням операторів динамічного розподілу даних;

– перевантажувати операцій для типів користувача;

– при розробці програм будувати об'єкти використовуючи успадкування та механізм віртуальних функцій;

– розробляти шаблони функцій та шаблони класів;

– при розробці програм вміти користуватися механізмом обробки виключних ситуацій;

– використовувати стандартну бібліотеку шаблонів соми С++.

Пререквізити: Дисципліна «*Програмування Ч.2. Об'єктно-орієнтоване програмування»* може вивчатись студентами після набуття окремих компетентностей на дисциплінах «Програмування Ч.1. Алгоритмізація та програмування» , «Інформатика».

Перелік предметних компетентностей здобувача вищої освіти:

Загальні компетентності (ЗК)

ЗК3 Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК4 Здатність застосовувати знання у практичних ситуаціях.

ЗК5 Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК7 Здатність працювати в команді.

ЗК8 Здатність вчитися і оволодівати сучасними знаннями.

Спеціальні компетентності (СК)

СК9. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

СК10 Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати прийняті рішення.

СК15 Здатність проектувати, розробляти та обслуговувати webзастосунки з динамічним контентом, використовуючи webтехнології, технології комп'ютерної графіки та анімації.

СК16. Здатність генерувати нові ідеї (креативність).

СК17 Здатність до абстрактного мислення, аналізу і синтезу.

Програмні результати навчання:

РН10 Здійснювати пошук інформації з різних джерел для розв'язання задач комп'ютерної інженерії.

РН11. Ідентифікувати, класифікувати та описувати роботу програмнотехнічних засобів комп'ютерної інженерії.

РН18. Системно мислити та застосовувати творчі здібності до формування нових та нестандартних рішень при розв'язуванні задач комп'ютерної інженерії.

РН20. Вміти ефективно працювати як індивідуально, так і у складі команди при вирішенні технічних та організаційних задач у професійній діяльності.

2. Опис навчальної дисципліни

2.1. Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин						Вид підсумкового контролю
			кредитів	годин	модулів	лекції	семінарські	практичні	лабораторні	самостійна робота	підготовка до екзамену	
Денна	3	5,6	7	210	4	60	-	-	60	60	30	екзамен

2.2. Дидактична карта навчальної дисципліни

№ заняття	Назва розділу (модуля), теми (змістового модуля), теми занять	Кількість годин							Навч.-метод. література (вказати № за п., сторінки)	Форми та засоби контролю
		ВСЬОГО	у тому числі:							
			лекції	семінарські	практичні	лабораторні	прак. інд.	с.р.с		
I семестр										
1	*Тема 1.1 Засоби робо- ти з бітами в С, С++та С++11-21. Базові нововведення в С++.*	12	4			4		4	[1], [9], [10], [13], [29],	Усне опит., тести
2	*Тема 1.2 Потоки даних С++. Стандартні та файлові.*	12	4			4		4	[2], [3], [12], [13], [15],	Усне опит., с.р.
3	*Тема 1.3 Базові функціональні відмінності мови С++ від С. Нововведення в С++.*	12	4			4		4	[1], [6], 7], [20], [22],	Усне опит., тести
4	*Тема 1.4 Концепція об'єктно-орієнтоване програмування.*	12	4			4		4	[26], [22]	Усне опит., с.р.
5	*Тема 1.5 Концепція класів ООП в мові С++*	16	4	-	-	6	-	6	[1],[4],[]5 [9], [11], [21],[23] [29],	Тести
6	*Тема 1.6 Дані та функції класів (члени класу).*	7	3	-	-	2	-	2	[2], [3], [12], [13], [15],[20],[26]	Усне опитування
7	*Тема 1.7 Статичні та сталі члени класів. Вбудовані та локальні класи.*	7	3	-	-	2	-	2	[1],[4], [11], [13],[21],[23] [29],	Тести
8	*Тема 1.8 Дружні функції та класи. Перевантаження операцій*	12	4	-	-	4	-	4	[1], [9], [10], [13], [29],	Усне опитування
	Всього годин за I семестр	**90**	**30**			**30**		**30**		

	II семестр									
9	*Тема 2.1 Зв'язки між об'єктами в програму ванні. Спадкування.*	8	3	-	-	3	-	2	[2], [3], [12], [13], [15],	Тести
10	*Тема 2.2 Дані та функ ції при успадкуванні. Множинне спадкуван- ня.*	6	2	-		2		2	[1], [6], 7], [20], [22],	Усне опит., тести
11	*Тема 2.3 Поліморфізм та пізнє зв'язування.Віртуальні функції. Абстрактні класи.*	10	3			3		4	[1], [9], [10], [13], [29],	Усне опитування
12	*Тема 2.4 Вказівники. Вказівник на функцію.Розумні вказівники.*	6	2			2		2	[1], [9], [10], [13], [29],	Усне опитування
13	*Тема 2.5 Потоки даних. Бібліотека потоків. Класи потоків*	10	3			3		4	[1], [9], [10], [13], [29],	Усне опит., тести.
14	*Тема 2.6 Шаблони функцій. Шаблони класів.*	8	3			3		2	[2], [3], [12], [13], [15],	Усне опитування
15	*Тема 2.7 Обробка виняткових ситуації.*	6	2			2		2	[1],[4], [11], [13],[21],[23] [29],	Усне опитування
16	*Тема 2.8 Перетворенення типів. RTTI.*	6	2			2		2	[1],[4], [11], [13],[21],[23] [29],	Усне опитування
17	*Тема 2.9 Стандартні контейнери. Стандартна бібліотека шаблонів мови С++.*	10	3			3		4	[1], [9], [10], [13], [29],	Усне опитування
18	*Тема 2.10 Спеціальні контейнери.*	8	3			3		2	1], [9], [10], [13], [29],	Усне опитування
19	*Тема 2.11 Нововведен- ня. С++11-С++20. Лямбда-вирази.*	6	2			2		2	[1],[4], [11], [13],[21],[23] [29],	Усне опитування
20	*Тема 2.12 Потоки задач. Паралельні обчислення.*	6	2			2		2	[1], [9], [10], [13], [29],	Усне опитування
	Всього за II семестр	**90**	**30**			**30**		**30**		
Всього годин		**180**	**60**	**-**	**-**	**60**	**-**	**60**		**ЕКЗАМЕН**

3. Теми лекційних занять

№ з/п	Назва теми	Кількість годин
1	***Тема 1.1 Засоби роботи з бітами в С, С++та С++11-21. Базові нововведення в С++.*** План 1. Основні побітові операції в мовах С та С++. 2. Використання побітових масок для обробки даних. 3. Побітові структури та операції над прапорцями. 4. Нововведення в С++11–21, що впливають на роботу з бітами. 5. Клас `bitset` у стандартній бібліотеці С++. **6.** Практичні приклади оптимізації коду за допомогою побітових операцій.	4
2	***Тема 1.2 Потоки даних С++. Стандартні та файлові.*** План 1. Потоки введення/виведення у мові С++. 2. Класи `istream`, `ostream`, `ifstream`, `ofstream`, `fstream`. 3. Читання та запис даних у текстові та бінарні файли. 4. Маніпулятори форматованого виведення (`setw`, `setprecision`, тощо). **5.** Обробка помилок у файлових потоках. **6.** Приклади створення програм із використанням потоків	4
3	***Тема 1.3 Базові функціональні відмінності мови С++ від С. Нововведення в С++.*** План 1. Ключові відмінності структурного програмування (С) та об'єктно-орієнтованого (С++). 2. Нові типи даних і розширення синтаксису в С++. 3. Простори імен (namespace) і модульність коду. 4. Автоматичне керування пам'яттю (RAII, smart pointers). 5. Нові можливості С++11–20 (auto, range-based for, constexpr). **6.** Приклади покращеного стилю програмування у С++.	4
4	***Тема 1.4 Концепція об'єктно-орієнтоване програмування.*** План 1. Основні принципи ООП: інкапсуляція, спадкування, поліморфізм. 2. Переваги ООП над процедурним підходом. 3. Об'єкти та класи як основні елементи ООП. 4. Абстракція та моделювання реальних об'єктів. **5.** Приклади застосування ООП у С++. **6.** Вплив ООП на архітектуру програмного забезпечення.	4
5	***Тема 1.5 Концепція класів ООП в мові С++*** План 1. Створення класів та об'єктів у С++. 2. Конструктори та деструктори. 3. Модифікатори доступу: `public`, `private`, `protected`. 4. Інкапсуляція та приховування даних. **5.** Методи класів і їх перевантаження. **6.** Практичні приклади побудови класів	4
6	***Тема 1.6 Дані та функції класів (члени класу).*** 1. Поля та методи класів. 2. Визначення функцій-членів поза класом. 3. Конструктори копіювання та переміщення.	3

	4. Передача об’єктів як аргументів функцій. **5.** Практичне застосування членів класу	
7	***Тема 1.7 Статичні та сталі члени класів. Вбудовані та локальні класи.*** План 1. Статичні змінні та функції у класі. 2. Сталі члени класів (`const`, `constexpr`). 3. Використання вбудованих класів. 4. Локальні класи та області їх застосування. 5. Приклади комбінування статичних та вбудованих елементів.	3
8	***Тема 1.8 Дружні функції та класи. Перевантаження операцій*** План 1. Призначення дружніх функцій та класів. 2. Правила доступу до приватних членів через `friend`. 3. Перевантаження арифметичних і логічних операторів. 4. Перевантаження операторів введення/виведення (`<<`, `>>`). 5. Обмеження перевантаження операторів. **6.** Практичні приклади реалізації перевантаження.	4
9	***Тема 2.1 Зв'язки між об'єктами в програму ванні. Спадкування.*** План 1. Основи концепції спадкування. 2. Типи спадкування (`public`, `protected`, `private`). 3. Базовий і похідний класи. **4.** Переваги повторного використання коду через спадкування.	3
10	***Тема 2.2 Дані та функції при успадкуванні. Множинне спадкування.*** План 1. Передача даних і методів у похідні класи. 2. Проблема діаманта при множинному спадкуванні. 3. Використання віртуального спадкування. **4.** Приклади реалізації множинного спадкування.	2
11	***Тема 2.3 Поліморфізм та пізнє зв’язування .Віртуальні функції. Абстрактні класи.*** План 1. Поняття поліморфізму. 2. Віртуальні функції та таблиця віртуальних викликів (vtable). 3. Абстрактні класи та чисті віртуальні функції. 4. Механізм пізнього зв’язування. 5. Приклади реалізації поліморфізму в C++. **6.** Практичне використання абстрактних класів.	3
12	***Тема 2.4 Вказівники. Вказівник на функцію .Розумні вказівники.*** План 1. Базові поняття вказівників у C++. 2. Вказівники на функції та їх використання. 3. Розумні вказівники (`unique_ptr`, `shared_ptr`, `weak_ptr`). **4.** Автоматичне керування пам’яттю.	2
13	***Тема 2.5 Потоки даних. Бібліотека потоків. Класи потоків*** План 1. Основи багатопоточності в C++. 2. Бібліотека `<thread>` та створення потоків. 3. Синхронізація потоків (`mutex`, `lock_guard`). **4.** Приклади паралельного виконання завдань.	3

14	***Тема 2.6 Шаблони функцій. Шаблони класів.*** План 1. Поняття шаблонів у C++. 2. Створення шаблонних функцій. 3. Шаблонні класи та параметризація типів. **4.** Переваги узагальненого програмування.	3
15	***Тема 2.7 Обробка виняткових ситуації.*** План 1. Концепція винятків у C++. 2. Ключові слова `try`, `throw`, `catch`. 3. Ієрархія стандартних винятків. 4. Створення власних типів винятків. 5. Обробка помилок у файлових і мережевих програмах. **6.** Безпечне завершення програм із винятками.	2
16	***Тема 2.8 Перетворенення типів. RTTI.*** План 1. Неявні та явні перетворення типів. 2. Оператори приведення типів (`static_cast`, `dynamic_cast`, `reinterpret_cast`, `const_cast`). **3.** Механізм RTTI (Run-Time Type Information). **4.** Приклади використання RTTI у спадкових ієрархіях.	2
17	***Тема 2.9 Стандартні контейнери. Стандартна бібліотека шаблонів мови C++.*** План 1. STL — структура та компоненти. 2. Контейнери `vector`, `list`, `deque`, `map`, `set`. 3. Ітератори та алгоритми STL. 4. Використання лямбда-функцій у STL. 5. Практичні приклади маніпулювання контейнерами. **6.** Переваги та недоліки використання STL.	3
18	***Тема 2.10 Спеціальні контейнери.*** План 1. Огляд спеціальних контейнерів (`unordered_map`, `unordered_set`, `priority_queue`). 2. Особливості реалізації хеш-контейнерів. 3. Використання черг, стеків і двосторонніх черг. **4.** Застосування спеціальних контейнерів у реальних задачах.	3
19	***Тема 2.11 Нововведення. C++11-C++20. Лямбда-вирази.*** План 1. Основні зміни стандартів C++11–20. 2. Поняття лямбда-виразів. 3. Захоплення змінних у лямбдах. **4.** Приклади використання лямбда-функцій у алгоритмах STL.	2
20	***Тема 2.12 Потоки задач. Паралельні обчислення.*** План 1. Концепція асинхронного програмування. 2. Класи `async`, `future`, `promise`. **3.** Створення та виконання потоків задач. **4.** Приклади паралельних обчислень у C++.	2
	Всього	**60**

4. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	*Тема 1.1 Засоби робо- ти з бітами в С, С++та С++11-21. Базові нововведення в С++.*	4
2	*Тема 1.2 Потоки даних С++. Стандартні та файлові.*	4
3	*Тема 1.3 Базові функціональні відмінності мови С++ від С. Нововведення в С++.*	4
4	*Тема 1.4 Концепція об'єктно-орієнтоване програмування.*	4
5	*Тема 1.5 Концепція класів ООП в мові С++*	6
6	*Тема 1.6 Дані та функції класів (члени класу).*	2
7	*Тема 1.7 Статичні та сталі члени класів. Вбудовані та локальні класи.*	2
8	*Тема 1.8 Дружні функції та класи. Перевантаження операцій*	4
9	*Тема 2.1 Зв'язки між об'єктами в програму ванні. Спадкування.*	3
10	*Тема 2.2 Дані та функції при успадкуванні. Множинне спадкуван ня.*	2
11	*Тема 2.3 Поліморфізм та пізнє зв'язування .Віртуальні функції. Абстрактні класи.*	3
12	*Тема 2.4 Вказівники. Вказівник на функцію .Розумні вказівники.*	2
13	*Тема 2.5 Потоки даних. Бібліотека потоків. Класи потоків*	3
14	*Тема 2.6 Шаблони функцій. Шаблони класів.*	3
15	*Тема 2.7 Обробка виняткових ситуації.*	2
16	*Тема 2.8 Перетворення типів. RTTI.*	2
17	*Тема 2.9 Стандартні контейнери. Стандартна бібліотека шаблонів мови С++.*	3
18	*Тема 2.10 Спеціальні контейнери.*	3
19	*Тема 2.11 Нововведення. С++11-С++20. Лямбда-вирази.*	2
20	*Тема 2.12 Потоки задач. Паралельні обчислення.*	2
Всього		**60**

6. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	*Тема 1.1 Засоби робо- ти з бітами в С, С++та С++11-21. Базові нововведення в С++.*	4
2	*Тема 1.2 Потоки даних С++. Стандартні та файлові.*	4
3	*Тема 1.3 Базові функціональні відмінності мови С++ від С. Нововведення в С++.*	4
4	*Тема 1.4 Концепція об'єктно-орієнтоване програмування.*	4
5	*Тема 1.5 Концепція класів ООП в мові С++*	6
6	*Тема 1.6 Дані та функції класів (члени класу).*	2
7	*Тема 1.7 Статичні та сталі члени класів. Вбудовані та локальні класи.*	2
8	*Тема 1.8 Дружні функції та класи. Перевантаження операцій*	4
9	*Тема 2.1 Зв'язки між об'єктами в програму ванні. Спадкування.*	2
10	*Тема 2.2 Дані та функції при успадкуванні. Множинне спадкування.*	2
11	*Тема 2.3 Поліморфізм та пізнє зв'язування .Віртуальні функції. Абстрактні класи.*	4
12	*Тема 2.4 Вказівники. Вказівник на функцію .Розумні вказівники.*	2
13	*Тема 2.5 Потоки даних. Бібліотека потоків. Класи потоків*	4
14	*Тема 2.6 Шаблони функцій. Шаблони класів.*	2
15	*Тема 2.7 Обробка виняткових ситуації.*	2
16	*Тема 2.8 Перетворення типів. RTTI.*	2
17	*Тема 2.9 Стандартні контейнери. Стандартна бібліотека шаблонів мови С++.*	4
18	*Тема 2.10 Спеціальні контейнери.*	2
19	*Тема 2.11 Нововведення. С++11-С++20. Лямбда-вирази.*	2
20	*Тема 2.12 Потоки задач. Паралельні обчислення.*	2
Всього		**60**

7. Форми та методи навчання

Форми навчання – це лекції-візуалізації (із застосуванням комп'ютерної техніки), проблемні лекції, практичні заняття (з використанням підходів «переверну- того класу» та виконанням завдань у мікрогрупах), лабораторні заняття, інтегровані заняття, заняття з використанням систем електронного навчання Moodle/classroom.github.com/classrooms; індивідуальні та групові консультації, самостійна робота (індивідуальна та в мікрогрупах під керівництвом викладача); використання елементів дистанційного навчання (за потреби): відеолекції, відео заняття і відеоконференції засобами Google Meet, Zoom тощо.

Підходи до навчання – використовуються студентоцентрований, проблемно-орієнтований, діяльнісний, комунікативний, професійно-орієнтований, міждисциплінарний підходи.

Для викладання навчальної дисципліни використовуються наступні ***методи навчання:***

- ☑ *пояснювально-ілюстративні* (спрямовані на повідомлення готової інформації різними засобами (словесними, наочними, практичними) та усвідомлення і запам'ятовування цієї інформації студентами);
- ☑ *компетентнісний* (навчання, спрямоване на розвиток навичок, умінь і якостей, які знадобляться в професійній діяльності);
- ☑ *репродуктивний* (використовується під час практичних і лабораторних занять, а також під час самостійної роботи студентів; передбачає роботу студентів за визначеним алгоритмом);
- ☑ *частково-пошукові або евристичні* (організація активного пошуку розв'язання поставлених або самостійно сформульованих пізнавальних завдань, над якими студенти працюють самостійно під керівництвом педагога або на основі евристичних програм та вказівок);
- ☑ *проєктно-дослідницькі* (групи студентів отримують комплекс завдань чи проблемне питання, визначений час для виконання; метод спрямований на розвиток пошукових, аналітичних якостей студентів, а також навичок командної роботи).

8. Розподіл балів, які отримують студенти

№ з/п	Назва теми	Кількість годин
1	*Тема 1.1 Засоби робо- ти з бітами в С, С++та С++11-21. Базові нововведення в С++.*	4
2	*Тема 1.2 Потоки даних С++. Стандартні та файлові.*	4
3	*Тема 1.3 Базові функціональні відмінності мови С++ від С. Нововведення в С++.*	4
4	*Тема 1.4 Концепція об'єктно-орієнтоване програмування.*	4
5	*Тема 1.5 Концепція класів ООП в мові С++*	6
6	*Тема 1.6 Дані та функції класів (члени класу).*	2
7	*Тема 1.7 Статичні та сталі члени класів. Вбудовані та локальні класи.*	2
8	*Тема 1.8 Дружні функції та класи. Перевантаження операцій*	4
9	*Тема 2.1 Зв'язки між об'єктами в програму ванні. Спадкування.*	4
10	*Тема 2.2 Дані та функції при успадкуванні. Множинне спадкування.*	4
11	*Тема 2.3 Поліморфізм та пізнє зв'язування .Віртуальні функції. Абстрактні класи.*	4
12	*Тема 2.4 Вказівники. Вказівник на функцію .Розумні вказівники.*	4
13	*Тема 2.5 Потоки даних. Бібліотека потоків. Класи потоків*	4
14	*Тема 2.6 Шаблони функцій. Шаблони класів.*	4
15	*Тема 2.7 Обробка виняткових ситуації.*	4
16	*Тема 2.8 Перетворення типів. RTTI.*	4
17	*Тема 2.9 Стандартні контейнери. Стандартна бібліотека шаблонів мови С++.*	4
18	*Тема 2.10 Спеціальні контейнери.*	3
19	*Тема 2.11 Нововведення. С++11-С++20. Лямбда-вирази.*	3
20	*Тема 2.12 Потоки задач. Паралельні обчислення.*	3
Всього		**50**

9. Критерії оцінювання результатів навчання з навчальної дисципліни з використанням багаторівневості

Контроль та оцінювання результатів навчання здійснюється згідно Положення про контроль і систему оцінювання результатів навчання здобувачів фахової передвищої освіти у ВСП «Фаховий Коледж ЧНУ».

Поточний модульний контроль (ПМК) здійснюється під час проведення лекційних, практичних та індивідуально-консультативних занять з метою перевірки рівня засвоєння теоретичних знань та практичних навичок студента. ПМК проводиться у формі написання письмових робіт, проміжних тестувань та активності й влучності обговорення відповідних тем під час навчальних занять. Згідно з навчальним планом семестровий контроль з дисципліни «*Програмування Ч.2. Об'єктно-орієнтоване програмування»* відбувається у формі екзамену.

Відвідування занять. Відсутність на аудиторному занятті не передбачає нарахування штрафних балів, оскільки фінальний рейтинговий бал студента формується виключно на основі оцінювання результатів навчання. Разом з тим, обговорення результатів виконання тематичних завдань, а також презентація / публічний виступ та участь у обговореннях та доповнення на практичних заняттях оцінюватимуться під час аудиторних занять.

Оцінювання пропущених контрольних заходів. Кожен студент має право відпрацювати пропущені з поважної причини (лікарняний, мобільність тощо) заняття за рахунок самостійної роботи.

Процедура оскарження результатів контрольних заходів оцінювання. Студент може підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами Положенням про апеляцію на результати підсумкового семестрового контролю знань студентів ВСП «Фаховий Коледж ЧНУ».

Академічна доброчесність. Політика та принципи академічної доброчесності визначені Етичним кодексом Чернівецького національного університету імені Юрія Федьковича (https://www.chnu.edu.ua/media/xe1lulcg/etychnyi-kodeks-chernivetskoho-natsionalnohouniversytetu.pdf).

Інклюзивне навчання. Засвоєння знань та умінь в ході вивчення дисципліни «*Програмування Ч.2. Об'єктно-орієнтоване програмування»* може бути доступним для більшості осіб з особливими освітніми потребами, окрім здобувачів з серйозними вадами зору, які не дозволяють виконувати завдання за допомогою персональних комп'ютерів, ноутбуків та/або інших технічних засобів.

Навчання іноземною мовою. У ході виконання завдань студентам може бути рекомендовано звернутися до англомовних джерел.

Підготовка до практичних занять та контрольних заходів здійснюється під час самостійної роботи студентів з можливістю консультування з викладачем у визначений час консультацій або за допомогою електронного листування (електронна пошта, месенджери).

Неформальна освіта. Регулюється Положенням про взаємодію формальної та неформальної освіти (https://www.chnu.edu.ua/media/3aykf41y/polozhennia-provzaiemodiiu-formalnoi-ta-neformalnoi-osvity.pdf). При наявності сертифікатів про проходження професійно-спрямованих психолого-педагогічних курсів, тренінгів, майстер-класів з неформальної освіти, участь у неформальній освіті студентам може бути перезараховано до 35% змістового матеріалу, що відповідає прослуханому матеріалу, за умови підготовки ними презентацій та нотаток за матеріалами прослуханого курсу, чи веб-заходу та їх публічного захисту на практичних заняттях.

Результати неформальної освіти можуть бути зараховані студенту згідно з Положенням ЧНУ "Про взаємодію формальної та неформальної освіти, визнання результатів навчання (здобутих шляхом неформальної та/або інформальної освіти, в системі формальної освіти)".

Якщо у студента наявні сертифікати про неформальну освіту з тематикою лабораторних робіт та лекцій, то студентові зараховуються бали за виконання лабораторних робіт, які покриваються пройденим курсом. Такий студент має лише виконати контрольну роботу та пройти підсумковий контроль у формі співбесіди студента з лектором.

Також, як виконані види роботи з відповідних тем, студенту можуть бути зараховані бали за наукові публікації у матеріалах науково-практичних конференцій
та фахових чи апробаційних виданнях або подання роботи на конкурс студентських наукових робіт.

Також, як можливості неформальної освіти студентам під час вивчення курсу ««*Програмування Ч.2. Об'єктно-орієнтоване програмування»* пропонується проходження курсів з отриманням сертифікатів, як індивідуальне завдання (ІНДЗ). На один модуль не більше 10 балів при 100% (9б – 80-99%, 8б – 60-79% тощо) проходженні курсів на віртуальних платформах (Prometheus, Coursera та інші).

Оцінювання знань студентів здійснюється за 100-бальною шкалою. Результати роботи студентів впродовж навчального семестру оцінюються в ході поточного контролю в діапазоні від 1 до 60 балів (включно), а результати підсумкового контролю (іспиту) оцінюються від 1 до 40 балів (включно).

Поточний контроль роботи студентів з навчальної дисципліни "Об'єктно- орієнтоване програмування" здійснюється за наступними критеріями:

- ☑ виконання та захист лабораторних робіт – 50 балів;
- ☑ тести в системах електронного навчання Moodle – 10 балів;
- ☑ іспит (теоретичні питання, здача та тести) – 40балів;

Усі лабораторні роботи виконуються у вигляді індивідуальних проєктів із синхронізацією на classroom.github.com/classrooms(викладача). Деякі лабораторні роботи виконуються в команді. На кожну роботу в команді призначається лідер команди, який розподіляє та координує роботу членів команди (на кожну наступ- ну роботу лідер в команді

змінюється), призначається репозиторій на github.com. Кожна команда формує звіт про виконання лабораторних робіт, подає на перевірку викладачу і захищає у вигляді презентації, коментарів (commit) на github.com та співбесіди з викладачем. При формуванні оцінки кожного студента за лабораторну роботу враховується: взаємне оцінювання членів команди (до 20% від загальної кількості балів), оцінка за програмну реалізацію та презентацію виконаного проєкту (оцінка самостійної роботи – до 30% від загальної кількості балів) та оцінка, отримана в результаті співбесіди з викладачем (до 20% від загальної кількості балів). Контрольна робота оцінюється згідно з критеріями оцінювання контрольних робіт.

Для успішного проходження підсумкового оцінювання здобувач освіти має набрати не менше 30% балів від заданої кількості балів за кожну лабораторну та тести в системі електронного навчання Moodle не менше 30%. Ті студенти, які зарезультатами поточного контролю отримали не менше 20 балів, допускаються до іспиту.

Кожен екзаменаційний білет (**40** балів) складається з двох теоретичних питань, задачі та посилання на тести, які знаходяться системі електронного навчання Moodle. За кожне теоретичне питання студент може отримати максимально **9** балів, за практичне завдання – **12** балів та тести **10** балів. У сумі з модульними контролями (**60** балів) це загалом складатиме максимально **100** балів.

Критерії оцінювання результатів навчання з навчальної дисципліни

Екзамен виставляється, як загальна підсумкова оцінка з дисципліни, яка складається з суми балів за результатами ПМК. Якщо студент на екзамені отримав незадовільну оцінку, то це вважається як академічна заборгованість і набрані бали не заносяться до відомості. За графіком деканату студент перескладає залік і його результати заносяться до окремої відомості.

Оцінюючи вирішення лабораторних завдань, викладач керується такими критеріями:

- повнота та послідовність виконання роботи
- правильність виконання і функціонування лабораторної роботи
- рівень вмінь застосовувати знання для аналізу практичних завдань чи проблемних питань,
- можливість поєднання теоретичного матеріалу в реальній проблемній ситуації.

Критерії оцінювання тестових багаторівневих завдань:

- 90-100 % правильних відповідей – "відмінно" (5);
- 70-89 % правильних відповідей – "добре" (4);
- 50-69 % правильних відповідей – "задовільно" (3);
- 49 % і менше правильних відповідей – "незадовільно" (2).

Шкала оцінювання результатів навчальних досягнень при вивченні дисципліни

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

Критерії оцінювання екзамену з дисципліни «*Програмування Ч.2. Об'єктно-орієнтоване програмування»* з використанням багаторівневості

- До екзамену допускаються студенти, які не мають академічної заборгованості, засвоїли навчальну програму з основ психології, регулярно відвідували лекції та семінарські заняття, виконували обов'язкові завдання на заняттях (ситуаційні задачі, тести), відвідували консультації з даної дисципліни.
- Рівень знань студентів оцінюється за двома критеріями: "відмінно", "добре", "задовільно" і "незадовільно".
- **«Відмінно»** - студент дав вичерпну відповідь на поставлені питання, показав глибокі ґрунтовні знання з усіх питань, вміє нестандартно мислити і давати оригінальне тлумачення проблем, здатний самостійно аналізувати, узагальнювати, робити висновки на основі аналізу конкретного матеріалу.
- **«Добре»** - студент правильно і досить повно виклав основні теоретичні питання, добре знає програмний матеріал, показав розуміння реальної дійсності. Відповідь ґрунтується на знанні основної літератури з дисципліни.
- **«Задовільно»** - студент правильно репродуктивно висвітлив питання на основі вивчення головним чином підручника і записів лекцій або при доброму знанні одних питань, відносно слабо знає інші питання.
- **«Незадовільно»** - студент не може викласти суть питання, не знає певних інформаційних категорій, не розуміє реальних процесів і явищ.

10. Форми поточного та підсумкового контролю

Методи контролю

У процесі оцінювання навчальних досягнень студентів застосовуються такі методи:

- Методи усного контролю: індивідуальне опитування, фронтальне опитування.
- Методи письмового контролю: письмове тестування, самостійні роботи, модульні контрольні роботи, есе.
- Методи практичного контролю: виконання лабораторних завдань.

Види контролю

Поточний контроль здійснюється на всіх етапах навчальної діяльності у формах: опитування студентів на предмет засвоєння навчального матеріалу; тестового контролю; виконання практичних завдань, письмових робіт.

Підсумковий контроль проводиться у кінці семестру у формі усного екзамену та заліку.

Підсумкове оцінювання за результатами вивчення дисципліни проводиться шляхом зваженого додавання результатів модульного контролю та результатів усного заліку.

Система оцінювання

Навчальні досягнення студентів із дисципліни «*Програмування Ч.2. Об'єктно-орієнтоване програмування»* оцінюються 100-бальною шкалою.

Кожний модуль включає бали за поточну роботу студента на семінарських (практичних) заняттях, виконання самостійних робіт, модульну контрольну роботу.

Контроль успішності студентів з урахуванням поточного і підсумкового оцінювання здійснюється відповідно до розподілу балів

Види та форми контролю. Поточний контроль за роботою студентів під час вивчення навчальної дисципліни здійснюється за допомогою наступних методів:

- ☑ захист студентами лабораторних робіт та проєктів;
- ☑ поточні опитування та тестування.

Проводиться під час лабораторних та практичних занять. Основне завдання – перевірка рівня підготовки студентів за визначеною темою (навчальним елементом). Основна мета – забезпечення зворотного зв'язку між викладачами та студентами, управління навчальною мотивацією студентів. Інформація, одержана при поточному контролі, використовується як викладачем – для коригування методів і засобів навчання, – так і студентами – для планування самостійної робо- ти.

Підсумковий контроль здійснюється у формі іспиту в терміни, встановлені графіком навчального процесу та в обсязі навчального матеріалу, визначеного робочою програмою дисципліни.

Засоби оцінювання:

- ☑ опитування теоретичного матеріалу;
- ☑ звіти з лабораторних роботи;
- ☑ індивідуальні та групові проєкти;
- ☑ презентації результатів виконаних завдань;
- ☑ тестові завдання.

1. Політика курсу

Політика щодо відвідування: відвідування занять є обов'язковим (виключення складають студенти, які навчаються за індивідуальним графіком та ті, кому зараховано результати неформальної освіти). Для осіб з особливими освітніми по- требами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей. Засвоєння теми лекції, пропущеної з поважної причини, перевіряється під час складання підсумкового контролю. Пропуск лекції з неповажної причини відпрацьовується студентом (співбесіда, реферат тощо). Пропущені практичні та лабораторні заняття, незалежно від причини пропуску, студент відпрацьовує згідно з графіком консультацій. За об'єктивних причин (наприклад, хвороба, участь у програмі міжнародного обміну, індивідуальний графік навчання) навчання може відбуватись у змішаній формі (очно-дистанційній) за погодженням із керівником курсу.

Політика академічної доброчесності: обов'язковими є посилання на джерела інформації у разі використання ідей, розробок, тверджень, відомостей; на- дання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використані методики досліджень і джерела інформації; списування під час контрольних заходів заборонені (в т. ч. із використанням мобільних пристроїв).

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ВСП «Фаховий коледж ЧНУ»:

- Етичний кодекс Відокремленого структурного підрозділу «Фаховий коледж Чернівецького національного університету імені Юрія Федьковича»
- Положенням про виявлення та запобігання академічному плагіату у Відокремленому структурному підрозділі «Фаховий коледж Чернівецького національного університеті імені Юрія Федьковича»
- Політика використання штучного інтелекту в Чернівецькому національному університету імені Юрія Федьковича

Використання здобувачами фахової передвишої 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 порушенням інших вимог академічної доброчесності.

Політика щодо дедлайнів та перескладання: роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (від -10% до

-50% від максимальної кількості балів – залежно від терміну затримки здачі робо- ти). Порушення терміну здачі роботи з поважної причини не призводить до втрати балів. Складання (перескладання) іспиту відбувається за встановленим деканатом розкладом. Якщо студента не допущено до складання іспиту через те, що він на- брав менше 20 балів протягом семестру, то до перескладання він має доздати викладачу лабораторні роботи.

Політика щодо оскарження оцінювання: забезпечення об’єктивності та прозорості оцінювання регламентується п.3.8-3.9 Положення ЧНУ "Про контроль і систему оцінювання результатів навчання здобувачів вищої освіти"; оскарження результатів підсумкового оцінювання здійснюється у відповідності до Положення ЧНУ "Про апеляцію на результати підсумкового семестрового контролю знань студентів".

12. Рекомендована література

1.1. Основна

1. Об’єктно-орієнтоване програмування. Мова С++: Навчальний посібник (матеріали лекцій) / Укл. Лазорик В.В., Фратавчан В.Г. Чернівці: Чернів. нац. ун-т ім. Ю. Федьковича, 2022, 115 с.
2. Фратавчан В.Г., Фратавчан Т.М., Лазорик В,В. Алгоритмізація та програмування, навчальний посібник для закладів вищої освіти. – ЧНУ, 2022, – 286 с.
3. **Stroustrup B.** A Tour of C++. Addison-Wesley. (3rd edition) Softcover and electronic versions, 2022. 190 p.
4. **Meyers S.** Effective modern C++: 42 specific ways to improve your use of C++11and C++14. O’Reilly Media, 2019. 334 p.
5. **Deitel P.,** Deitel H. C++20 for programmers, 3rd edition. Pearson Education Limited, 2021. 1008 p.
6. **Galowitz J**. C++17 STL Cookbook, Packt Publishing, 2018. 532 p.
7. **Josuttis N. M.** C++20 – The complete guide, Leanpub, 2021. 764 p.
8. **Roth S.** Clean C++ 20: Sustainable software development patterns and best practice, 2 nd edition. Apress, 2021. 508 p.
9. **Potter L**. Hands-on mobile and embedded development with Qt 5: Build apps for Android, iOS, and Raspberry Pi with C++ and Qt. Packt Publishing, 2019, 363 p.
10. **Stroustrup B.** The C++ Programming Language (Fourth Edition). Addison Wesley. Reading Mass, 2013. 1360 p. (1st edition 1985, 2nd edition 1991, 3rd edition 1997, "special" edition 2000, 4th edition 2013)
11. **Бублик В.В.** Об’єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТ- книга, 2015. – 624 с.: іл.
12. **Grady Booch , Robert A. Maksimchuk, Michael W. Engle , Dr. Bobbi Young, Jim Conallen** :Object-Oriented Analysis and Design with Applications (3rd Edition)/ Addison-Wesley Professional, 2007. 720 p.
13. **Stephen Prata** C++ Primer Plus (Developer's Library) Softcover, Addison-Wesley Professional, 2011. 1202 p.
14. **Stanley B. Lippman; Josée Lajoie; Barbara E. Moo** C++ Primer (5th ed.) , 2012. 972 p.
15. **Бєлов Ю. А.** Вступ до програмування мовою С++. Організація обчислень / Ю. А. Бєлов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. – К.: ВПЦ "Київський університет", 2012.
16. **Боровльова С. Ю.** Базовий С++ : навчальний посібник / С. Ю. Боровльова, А. В. Швед. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2017. – 116 с.
17. **Карнаух Т. О.** Вступ до програмування мовою С++. Організація даних : навчальний посібник / Т. О. Карнаух, Ю. В. Коваль, М. В. Потієнко, А. Б. Ставровський. – К.: Видавничо-поліграфічний центр "Київський університет", 2015. – 151 с.
18. **Грицюк Ю.І.** Об'єктно-орієнтоване програмування мовою С++ : навчальний посібник. . / Ю.І. Грицюк, Т.Є. Рак. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 404 с.

1.2. Допоміжна

1. Основи мов програмування С та С++. Частина 1 : Комп’ютерний практикум / Укл.:Лазорик В.В., Фратавчан В. Г. – Чернівці: ЧНУ, 2023. – 107 с.
2. **Васильєв О.** Програмування С++ в прикладах і задачах: навч. посібник / Васильєв О. – К. : Ліра К, 2019. 382 с.
3. **Трофименко О.Г**., Прокоп Ю.В., Задерейко О.В. Алгоритмізація та

програмування : навчально-методичний посібник. Одеса : Фенікс, 2020. 310 с. Веб-сайт. URL: http://dspace.onua.edu.ua/handle/11300/12345 (дата

звернення: 1.09.22)

4. **Грицюк Ю.І.** Програмування мовою С++ : навч. посібн. / Ю.І. Грицюк, Т.Є. Рак. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 292 с.
5. **Винник В. Ю.** Основи програмування мовою Сі++ .– Житомир, 2008.
6. **Deitel H. M., Deitel P.J.** : C++ How to Program / Pearson; 10th edition, 2016, 1080 p.
7. **Stroustrup B.** Programming -- Principles and Practice Using C++ (Second Edition). Addison-Wesley, Softcover and electronic versions, 2014. 1312 p. (1st Edition 2008, 2nd edition 2014)
8. **Herb Schildt's** C++ Programming Cookbook, Osborne/McGraw Hill, 2008. 509 pages
9. **Jesse Liberty, Rogers Cadenhead** Sams Teach Yourself C++ in 24 Hours (Sams Teach Yourself in 24 Hours) 5th Edition / Publisher : Sams; 5th edition, 2011. 444 p.
10. **Глинський Я.М.** С++ і С++ Builder / Я.М. Глинський, В.Є. Анохін, В.А. Ряжська. – Львів : Вид-во "Деол", СПД Глинський, 2003. – 192 с.
11. **Войтенко В.В. Морозов А.В.** Теорія та практика (мова С++).– Житомир, 2002.

1.3. Інформаційні ресурси

1. C++ reference : веб-сайт. URL: https://en.cppreference.com/w/ (дата звернення: 1.09.22)
2. cplusplus.com : веб-сайт. URL: https://cplusplus.com/ (дата звернення: 1.09.22)
3. Уроки програмування на С++ : веб-сайт. URL: https://acode.com.ua/uroki-po- cpp/ (дата звернення: 1.09.22)
4. News, Status & Discussion about Standard C++ : веб-сайт. URL: https://isocpp.org/ (дата звернення: 1.09.22)
5. Офіційні FAQ C++ : веб-сайт. URL: https://isocpp.org/wiki/faq.
6. C++ Annotations Version 12.2.0 Frank B. Brokken : веб-сайт. URL: http://www.icce.rug.nl/documents/cplusplus/ (дата звернення: 1.09.22)
7. C++ Core Guidelines : веб-сайт. URL: https://github.com/isocpp/CppCoreGuidelines/blob/master/CppCoreGuidelines.md (дата звернення: 1.09.22)
8. Lessons on the development of 64-bit C/C++ applications : веб-сайт. URL: https://pvs-studio.com/en/blog/lessons/ (дата звернення: 1.09.22)
9. C++ language documentation : веб-сайт. URL: https://learn.microsoft.com/en-us/cpp/cpp/?view=msvc-170
10. LearnCpp.com : веб-сайт. URL: https://www.learncpp.com/ (дата звернення: 1.09.22)
11. SimpleCode. Основи С++. Програмування для початківців : веб-сайт. URL: https://www.youtube.com/playlist?list=PLQOaTSbfxUtCrKs0nicOg2npJQYSPGO 9r . (дата звернення: 1.09.22)
12. Your Resource for C and C++ Programming : веб-сайт. URL: www.cprogramming.com (дата звернення: 1.09.22)
13. C++ Programming Language : веб-сайт. URL: https://www.geeksforgeeks.org/c- plus-plus/ (дата звернення: 1.09.22)
14. Вікіпедія: веб-сайт. URL: C++, C++ Standard Library, More C++ Idioms, C++11, C++03, TR1, Template (C++), Compatibility of C and C++, Template metaprogramming, C++ classes.