

**Чернівецький національний університет імені Юрія Федьковича**

(повне найменування закладу вищої освіти)

**Навчально-науковий інститут фізико-технічних і комп'ютерних наук**

(назва інституту/факультету)

**Кафедра комп'ютерних систем та мереж**

(назва кафедри)

**СИЛАБУС**

**навчальної дисципліни**

**OK14. Програмування. Ч2. Програмування мовою C++**

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

**обов'язкова**

(обов'язкова чи вибіркова)

**Освітньо-професійна програма «Комп'ютерна інженерія»,**

**“Програмування мобільних і вбудованих комп'ютерних систем та засобів Інтернету речей”**

**Спеціальність 123 – Комп'ютерна інженерія**

(шифр і назва спеціальності)

**Галузь знань 12 – Інформаційні технології**

(шифр і назва галузі знань)

**Рівень вищої освіти – перший (бакалаврський)**

(вказати: перший (бакалаврський)/другий (магістерський)/третій (освітньо-науковий))

**Навчально-науковий інститут фізико-технічних та комп'ютерних наук**

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

**Мова навчання – українська**

(мова, на якій читається дисципліна)

**Розробник: Танасюк Юлія Володимирівна, доцент кафедри КСМ,**

**кандидат фіз.-мат. наук, доцент**

(вказати авторів (викладач (ів)), їхні посади, наукові ступені, вчені звання)

**Кількість кредитів: 6**

**Форми навчальної діяльності: лекції, лабораторні роботи, самостійна робота**

**Форма підсумкового контролю: іспит**

**Профайл викладача (-ів)** <https://csn.chnu.edu.ua/employees/tanasyuk-yuliya-volodymyrivna/>

**Контактний тел.** +(38) 0372 50 94 32

**E-mail:** [y.tanasyuk@chnu.edu.ua](mailto:y.tanasyuk@chnu.edu.ua)

**Сторінка курсу в Moodle** <https://moodle.chnu.edu.ua/course/view.php?id=3485>

**Консультації** згідно з розкладом (один раз на тиждень)

## **1. Анотація дисципліни**

Навчальна дисципліна «Програмування. Ч2. Програмування мовою С++» спрямована на вивчення основ алгоритмізації та особливостей програмування прикладних і наукових задач мовою програмування високого рівня та набуття студентами практичних навичок зі структурного, процедурного й об'єктно-орієнтованого програмування, які допоможуть їм ефективно використовувати сучасні здобутки технологій програмування при опануванні інших фахових та спеціалізованих дисциплін.

**2. Мета навчальної дисципліни:** оволодіння студентами фундаментальних основ програмування мовою високого рівня С++ та набуття практичних навичок застосування базових принципів об'єктно-орієнтованого програмування та традиційного процедурного програмування для розв'язання широкого кола задач. Дисципліна містить відомості: про лексеми мови С++, основні та структуровані типи даних, базові конструкції (оператори розгалуження, циклу, переходу), вказівники та посилання, як засоби доступу до статичної та динамічної пам'яті, масиви та рядки, функції, структури, об'єднання, динамічні структури даних (стеки, черги, лінійні списки, бінарні дерева), класи та стандартну бібліотеку шаблонів (STL).

### **Завдання дисципліни:**

- ознайомлення із функціональними можливостями, що надаються мовою С++, та застосування їх на практиці для написання програм;
- використання основних принципів структурного програмування для розв'язання математичних задач та залучення процедурного підходу для створення багатофайлових проектів;
- засвоєння засобів об'єктно-орієнтованого програмування та використання їх на практиці для моделювання повнофункціональних програмних об'єктів.

## **3. Пререквізити**

Для успішного опанування матеріалів цієї дисципліни слухачам необхідно попередньо вивчити такі дисципліни: Програмування. Ч1. Основи алгоритмізації та програмування на Python і JavaScript, Вища математика, Комп'ютерна логіка. Ч1. Комп'ютерна арифметика, Комп'ютерна дискретна математика.

## **4. Результати навчання**

У результаті вивчення навчальної дисципліни студенти повинні:

### **4.1. Знати:**

- принципи створення програм із використанням базових конструкцій для розв'язання різного типу задач;
- формат визначення функцій, зокрема зі змінною кількістю параметрів, реалізацією механізму переважання та використання аргументів різних типів та способів їх передавання у функцію;

- особливості зберігання у пам'яті масивів, рядків, бітових послідовностей, принципи їх створення та оброблення;
- механізми доступу до об'єктів за значенням і за адресою;
- принципи визначення та використання структур і класів;
- можливості стандартної бібліотеки шаблонів STL та застосування таких контейнерів, як вектори, бітові рядки, множини, асоціативні масиви, алгоритми, потоки вводу/виводи.

#### **4.2. Вміти:**

- аналізувати і створювати програмні модулі із дотриманням принципів структурного, процедурного й об'єкто-орієнтованого програмування.
- виконувати формалізацію поставленої задачі та розробляти підходи до пошук рішень;
- розробляти алгоритм розв'язку типових прикладних та математичних задач різного рівня складності;
- використовувати основні прийоми програмної реалізації типових обчислювальних алгоритмів;
- створити консольний застосунок для реалізації розробленого алгоритму мовою C++;
- при написанні програм використовувати базові та складові структури даних;
- виконувати зчитування, пошук та збереження у файли;
- забезпечити правильність взаємодії користувачів із програмою;
- застосовувати ефективні способи зберігання, зчитування та оброблення інформації;
- налагоджувати і перевіряти правильність роботи програми з використанням засобів інтегрованого середовища програмування.

### **4.3. Набути компетентностей:**

#### *ЗК – загальних*

- ЗК1. Здатність до абстрактного мислення, аналізу і синтезу.
- ЗК2. Здатність вчитися і оволодівати сучасними знаннями.
- ЗК3. Здатність застосовувати знання у практичних ситуаціях.
- ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.
- ЗК6. Навички міжособистісної взаємодії.
- ЗК7. Вміння виявляти, ставити та вирішувати проблеми.
- ЗК8. Здатність працювати в команді.

#### *ФК – фахових (спеціальних)*

- ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
- ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
- ФК11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.
- ФК13. Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.
- ФК14. Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.
- ФК15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.
- ФК16. Здатність застосовувати технології комп'ютерних систем і мереж, дискретної обробки інформації та числових методів для реалізації інформаційно-вимірювальних систем і систем передачі даних.

#### *ПРН – програмних результатів навчання*

- 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.
- ПРН3. Знати новітні технології в галузі комп'ютерної інженерії.

ПРН4. Знати та розуміти вплив технічних рішень в суспільному, економічному, соціальному і екологічному контексті.

ПРН6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПРН8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

ПРН11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

ПРН12. Вміти ефективно працювати як індивідуально, так і у складі команди.

ПРН14. Вміти поєднувати теорію і практику, а також ухвалювати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням загальнолюдських цінностей, суспільних, державних та виробничих інтересів.

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

## 5. Опис навчальної дисципліни

### 5.1. Загальна інформація

Назва навчальної дисципліни <i>OK14. Програмування. Ч2. Програмування мовою C++</i>												
Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин						Вид підсумкового контролю
			кредитів	годин	змістових модулів	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	1	2	6	180	3	45	-	-	45	90	-	іспит
Заочна	1	2	6	180	3	12	-	-	8	160	-	іспит

**Примітка.** Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить: для денної форми навчання – 1 ((45+45)/90)  
для заочної форми навчання – 0.125 ((12+8)/160)

## 5.2. Дидактична картка навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	Денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Змістовий модуль 1. Типи даних та базові конструкції мови C++</b>												
Тема 1. Структура програми мовою C++. Лексеми	16	4	-	6	-	6	14	1	-	1	-	12
Тема 2. Типи даних	13	2	-	3	-	8	20	1	-	1	-	18
Тема 3. Позначки операцій та розділювачі	16	2	-	2	-	12	16	1	-	1	-	14
Тема 4. Основні оператори мови C++	24	4	-	6	-	14	22	1	-	1	-	20
Разом за змістовим модулем 1	69	12	-	17	-	40	72	4	-	4	-	64
<b>Змістовий модуль 2. Складові типи даних в C++</b>												
Тема 5. Вказівники та адресна арифметика	16	2	-	6	-	8	18	1	-	0	-	17
Тема 6. Масиви та рядки	30	6	-	12	-	12	22	2	-	1	-	19
Разом за змістовим модулем 2	46	8	-	18	-	20	40	3	-	1	-	36
<b>Змістовий модуль 3. Функції та структури. Основні принципи об'єктно-орієнтованого програмування. Класи.</b>												
Тема 7. Функції мови C++	21	6	-	5	-	10	14	1	-	1	-	12
Тема 8. Структури та об'єднання. Динамічні структури даних.	22	9	-	3	-	10	12	1	-	1	-	10
Тема 9. Клас, як абстрактний тип даних	22	10	-	2	-	10	42	3	-	1	-	38
Разом за змістовим модулем 3	65	25	-	10	-	30	68	5	-	3	-	60
<b>Усього годин</b>	<b>180</b>	<b>45</b>	<b>-</b>	<b>45</b>	<b>-</b>	<b>90</b>	<b>180</b>	<b>12</b>	<b>-</b>	<b>8</b>	<b>-</b>	<b>160</b>

### 5.3. Теми лабораторних занять

№	Назва теми	Кількість годин
1.	Форматне введення/виведення у мовах C/C++	3
2.	Лінійні програми на C++	4
3.	Порозрядні операції. Бітові поля	4
4.	Оператори мови C++	4
5.	Вказівники та одновимірні масиви	4
6.	Опрацювання двовимірних масивів	4
7.	Символьна інформація та рядки	4
8.	Функції користувача	4
9.	Структури	4
10.	Класи мови C++	10
	Разом	45

Програмне забезпечення для виконання лабораторних робіт: Microsoft Visual Studio 2019/202x Community, CLion, online-середовища C++ shell <https://cpp.sh/>, Replit <https://replit.com/languages/cpp>, Online GDB [https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

### 5.4. Самостійна робота

№	Назва теми	Кількість годин
1.	Налаштування компілятора. Попередження та повідомлення про помилки. Режими Debug і Release	4
2.	Пріоритет виконання операцій. Функції стандартної бібліотеки <code>cmath</code>	2
3.	Область видимості об'єктів програми. Класи пам'яті. Простір імен	4
4.	Директиви препроцесора. Файли-заголовки	4
5.	Функції для обробки символьної та рядкової інформації бібліотеки <code>string</code>	6
6.	Шаблони функцій	4
7.	Вказівники та посилання на функцію	4
8.	Типи даних, визначені користувачем. Перейменування типів ( <code>typedef</code> і <code>type alias</code> ). Класи-перелічування <code>enum</code>	3
9.	Оброблення помилок. <code>cerr</code> та <code>exit()</code> , <code>assert</code> і <code>static_assert</code>	4
10.	Лямбда-вирази	6
11.	Види конструкторів. Конструктори делегування. Деструктори. Статичні компоненти класу.	6
12.	Типи зв'язків між об'єктами класів: асоціація, композиція, агрегація, залежність	6
13.	Класи та шаблони. Створення сімейства класів.	3
14.	Робота з файлами. Методи обміну з потоками. Помилки потоків. Файлові потоки. Рядкові потоки	6
15.	Бінарні файли у мові C++. Рядковий ввід-вивід.	4

№	Назва теми	Кількість годин
16.	Множинне успадкування. Віртуальні базові класи. Динамічне приведення типів <code>dynamic_cast</code>	3
17.	Шаблони класів. Контейнерні класи.	4
18.	Обробка виняткових ситуацій: <code>try</code> і <code>catch</code>	4
19.	Розумні вказівники: <code>std::unique_ptr</code> , <code>std::shared_ptr</code> , <code>std::weak_ptr</code> . Посилання r-value. Переміщення. Функція <code>std::move()</code>	6
20.	Стандартна бібліотека C++. Вектори. Ітератори. Алгоритми.	8
	Разом	90

## 6. Форми і методи навчання

**Форми навчання:** це проблемні й оглядові лекції, лабораторні заняття, заняття із застосуванням комп'ютерної та телекомунікаційної техніки, інтерактивні заняття з навчанням одних студентів іншими, відеолекції, відеозаняття і відеоконференції засобами Google Meet, Cisco Webex, заняття з використанням системи електронного навчання Moodle та освітнього порталу Cisco Networking Academy [netacad.com](http://netacad.com).

**Методи:** проблемний виклад матеріалу, частково-пошукові та дослідницькі лабораторні практикуми, презентації, огляд практичних випадків реалізації та розв'язання завдань, консультації і дискусії, робота в інтернет-класі: електронні лекції, лабораторні роботи, тести, онлайн-консультації та ін., спрямовані на активізацію і стимулювання навчально-пізнавальної діяльності студентів.

**Підходи до навчання:** студенто-центрований, проблемно-орієнтований, діяльнісний, комунікативний, професійно-орієнтований, міждисциплінарний підходи.

**Реалізація навчального процесу** здійснюється під час лекційних, лабораторних занять, самостійної позааудиторної роботи з використанням сучасних інформаційних технологій навчання, консультацій з викладачами.

Для формувань умінь та навичок застосовуються такі **методи навчання:**

### 6.1. Словесні методи навчання. Навчальна лекція

За допомогою даного методу забезпечується усне викладення матеріалу великого обсягу й складності логічних побудов, доказів і узагальнень. У ході лекції використовуються прийоми усного викладення інформації, привертання уваги протягом тривалого часу, активізації мислення студентів, прийоми забезпечення логічного запам'ятовування, асоціативного мислення, пояснення, доведення, аргументування доказів, класифікації, систематизації та узагальнення. В залежності від специфіки лекційного матеріалу подеколи використовується лекція-діалог.



## **6.2. Індуктивний метод навчання**

Цей метод навчання використовується у рамках лекційних занять, коли матеріал носить, здебільшого, фактичний характер. Під час лабораторних занять метод застосовується при виконанні технічних завдань, коли студенти використовують раніше здобуті теоретичні знання при роботі з конкретними програмними продуктами та технічними засобами.

## **6.3. Репродуктивний метод навчання**

Цей метод навчання використовується у рамках лекційних і лабораторних занять, а також під час самостійної роботи студентів. Метод передбачає роботу студентів за визначеним алгоритмом. Згідно з методом для виконання завдань студентам надаються методичні вказівки, правила і навчальні приклади.

## **6.4. Проблемно-пошукові методи навчання**

Проблемно-пошукові методи застосовуються у ході проблемного навчання, а саме в процесі виконання лабораторних робіт та індивідуальних науково-дослідних завдань, де як проблемну ситуацію слід розглядати невідповідність між тим, що вивчається і вже засвоєним матеріалом. При використанні проблемно-пошукових методів навчання викладач використовує такі прийоми: створює проблемну ситуацію (ставить запитання, пропонує задачу, експериментальне завдання), організує колективне обговорення можливих підходів до рішення проблемної ситуації, стимулює висування гіпотез, генерування ідей, тощо. Студенти роблять припущення про шляхи вирішення проблемної ситуації, узагальнюють раніше набуті знання, виявляють причини явищ, пояснюють їхнє походження, обирають найбільш раціональний варіант розв'язання проблемної ситуації. Викладач обов'язково керує цим процесом на всіх етапах, зокрема за допомогою запитань-підказок. До того ж цей метод використовується при самостійному опрацюванні матеріалів в системі дистанційної освіти «Moodle».

## **6.5. Наочний метод навчання**

Наочний метод достатньо важливий для студентів, оскільки забезпечує візуальне подання навчального матеріалу, зокрема, з використанням інформаційно-комунікаційних технологій. При викладанні дисципліни наочний метод навчання поєднується зі словесними методами для представлення інформації у вигляді таблиць, рисунків, схем та діаграм.

## **7. Система контролю та оцінювання**

Засобами оцінювання та демонстрування результатів навчання є:

- завдання на лабораторні роботи;
- тестові завдання по темах;
- презентації результатів виконаних завдань;
- практичні завдання по варіантах.

Формами поточного контролю рівня знань є усна та письмова відповідь студента при захисті виконаних лабораторних робіт, підготовка звітів до лабораторних робіт, відповіді на контрольні запитання, складання тестів у системі дистанційної освіти,

виконання практичних завдань, а також письмова відповідь при написанні модульних контрольних робіт.

Формами підсумкового контролю рівня знань є усна та письмова відповідь студента при складанні іспиту.

### 7.1. Критерії оцінювання результатів навчання з навчальної дисципліни

Критерієм успішного проходження здобувачем освіти підсумкового оцінювання є досягнення ним мінімальних порогових рівнів оцінок за кожним запланованим завданням навчальної дисципліни.

У залежності від характеру відповіді студента кількість балів за кожен вид діяльності може бути визначена за такими критеріями:

К-ть балів	Критерії оцінки
Max	Студент дає вичерпну відповідь на поставлене запитання
0,8 · Max	Студент при відповіді на поставлене запитання припустився незначних неточностей, які не впливають на суть відповіді
0,6 · Max	Студент при відповіді на поставлене запитання припустився помилок, які виправляє за допомогою викладача; в середньому може дати правильні відповіді на 50% питань теми
0,4 · Max	Студент при відповіді на поставлене запитання припустився суттєвих помилок, які все ж таки виправляє за допомогою викладача; дає правильні відповіді на 30% питань теми
0,2 · Max	Студент за допомогою викладача фрагментарно відповідає на запитання, проте не в повній мірі володіє мінімальним рівнем знань з даного питання
0	Характер відповідей дає підставу стверджувати, що студент неправильно зрозумів суть питання чи не знав правильної відповіді, а тому відповідав, припускаючись грубих помилок.

### Шкала та критерії оцінювання: національна та ЄКТС (Європейська кредитна трансферно-накопичувальна система, ECTS)

Оцінка за шкалою ЄКТС	Пояснення	Оцінка за 100-бальною шкалою	Оцінка за національною шкалою
<b>A</b>	відмінно	<b>90 – 100</b>	<b>відмінно</b>
<b>B</b>	дуже добре	<b>80-89</b>	<b>добре</b>
<b>C</b>	добре	<b>70-79</b>	
<b>D</b>	задовільно	<b>60-69</b>	<b>задовільно</b>
<b>E</b>	достатньо	<b>50-59</b>	
<b>FX</b>	(незадовільно) з можливістю повторного складання	<b>35-49</b>	<b>незадовільно</b>
<b>F</b>	(незадовільно) з обов'язковим повторним вивченням курсу	<b>1-34</b>	

## Розподіл балів, які отримують студенти

Поточне тестування та індивідуальна робота													Підсумковий контроль (іспит)	Загальна сума балів
Змістовий модуль 1				Змістовий модуль 2				Змістовий модуль 3						
T1	T2	T3	T4	M1	T5	T6	T7	M2	T8	T9	T10	M3	40	100
3	4	3	4	8	5	3	5	5	5	5	5	5		

T1, T2, ... T10 – теми змістових модулів, M1, M2, M3 – модульні контрольні роботи

### 7.2. Перелік тем і розподіл максимально можливої кількості балів, які отримують студенти за виконання всіх видів навчальної діяльності

#### Змістовий модуль 1. Типи даних та базові конструкції мови C++

T1. Структура програми мовою C++. Лексеми.

1.1. Вступ до мови C++. Структура програми.

1.2. Склад мови C++. Лексеми. Константи.

Лабораторна робота № 1. Форматне введення/виведення у мовах C/C++ (3 бали)

T2. Типи даних.

Лабораторна робота № 2. Лінійні програми на C++ (4 бали)

T3. Типи операцій мови C++.

Лабораторна робота № 3. Порозрядні операції. Бітові поля (3 бали)

T4. Основні оператори мови C++.

4.1. Базові конструкції. Оператори розгалуження.

4.2. Оператори циклу та керування ходом програми.

Лабораторна робота № 4. Оператори мови C++ (4 бали)

M1. Модульна контрольна робота 1 (8 балів)

#### Змістовий модуль 2. Складові типи даних в C++

T5. Вказівники та адресна арифметика.

Лабораторна робота № 5. Вказівники та одновимірні масиви (5 балів)

T6. Масиви та рядки.

6.1. Одновимірні масиви.

6.2. Двовимірні масиви.

Лабораторна робота № 6. Опрацювання двовимірних масивів (3 балів)

T7. Символи та рядки

Лабораторна робота № 7. Символьна інформація та рядки (5 балів)

M2. Модульна контрольна робота 2 (5 балів)

#### Змістовий модуль 3. Функції та структури. Основні принципи об'єктно-орієнтованого програмування. Класи

T8. Функції мови C++.

8.1. Функції: формат визначення, опис, виклик.

8.2. Функції зі змінною кількістю параметрів. Перевантаження функцій.

8.3. Посилання. Способи передачі параметрів у функцію.

Лабораторна робота № 8. Функції користувача (5 балів)

T9. Структури та об'єднання. Динамічні структури даних.

9.1. Структури та об'єднання.

9.2. Динамічні структури даних: стеки, черги, списки, бінарні дерева

Лабораторна робота № 9. Структури (5 балів)

T10. Клас, як абстрактний тип даних.

10.1. Визначення класу. Компонентні дані і компонентні функції класу.

10.2. Принцип інкапсуляції. Статуси доступу компонент класу. Конструктори та деструктор.

10.3. Дружні функції класів. Перевантаження стандартних операцій.

10.4. Успадкування класів. Віртуальні базові класи. Множинне успадкування.

10.5. Віртуальні функції. Абстрактні класи. Поліморфізм.

10.6. Шаблони класів. Стандартна бібліотека шаблонів STL. Контейнери.

Лабораторна робота № 10. Класи мови C++ (5 балів)

M3. Модульна контрольна робота 3 (5 балів)

Підсумковий модуль-контроль – іспит (40 балів).

### 7.3. Умови зарахування результатів неформальної освіти

Студенти, згідно Положення ЧНУ «Про неформальну освіту» можуть отримати додаткові бали, або бути звільненими від окремих видів робіт за визначеними темами, якщо у них наявні сертифікати про неформальну освіту з проблем, які вивчаються в рамках дисципліни «Програмування. Ч2. Програмування мовою C++».

Також, як виконані види роботи з відповідних тем студентам зараховуються бали за наукові публікації у матеріалах науково-практичних конференцій та фахових чи апробаційних виданнях.

### 7.4. Політика курсу

Самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей).

**Академічна доброчесність:** посилення на джерела інформації у разі використання ідей, розробок, тверджень, відомостей; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використані методики досліджень і джерела інформації.

**Відвідування:** Відвідування занять є обов'язковим. Засвоєння пропущеної теми лекції 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.

## 8. Рекомендована література Фахова (основна)

1. Програмування мовою C++: навчальний посібник / укл. Танасюк Ю.В. Чернівці : ЧНУ, 2022. 208 с. (електронне видання).
2. Васильєв О. Програмування C++ в прикладах і задачах: навч. посібник : 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. URL : <http://dspace.onua.edu.ua/handle/11300/12345>
4. Prata S. C++ Premier Plus. Kindle Edition, 2018. 1440 p.
5. Strastrup B. Tour of C++, 2<sup>nd</sup> edition. Addison-Wesley Professional, 2018. 256 p.
6. Meyers S. Effective modern C++: 42 specific ways to improve your use of C++11 and C++14. O'Reilly Media, 2019. 334 p.
7. Deitel P., Deitel H. C++20 for programmers, 3<sup>rd</sup> edition. Pearson Education Limited, 2021. 1008 p.

### Допоміжна

1. Galowitz J. C++17 STL Cookbook. Packt Publishing, 2018. 532 p.
2. Josuttis N. M. C++20 – The complete guide. Leanpub, 2021. 764 p.
3. Roth S. Clean C++ 20: Sustainable software development patterns and best practice, 2<sup>nd</sup> edition. Apress, 2021. 508 p.
2. 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.
1. Eckel B. Thinking in C++. Vol. 1: Introduction in standard C++, 2<sup>nd</sup> edition. Prentice Hall, 2000. 840 p.
2. Eckel B. Allison Ch. Thinking in C++. Vol. 2: Practical programming, 2<sup>nd</sup> edition. Prentice Hall, 2003. 832 p.

## 9. Інформаційні ресурси

1. CPA: Programming Essentials in C++. – Cisco Networking Academy Course. URL : [netacad.com](https://netacad.com).
2. Основи програмування мовами C і C++. URL : [c++studio.com](https://c++studio.com)
3. Уроки програмування на C++. URL : <https://acode.com.ua/uroki-po-cpp/>
4. SimpleCode. Основи C++. Програмування для початківців: URL : <https://www.youtube.com/playlist?list=PLQOaTSbfxUtCrKs0nicOg2npJQYSPGO9r>
5. Your Resource for C and C++ Programming. URL : [www.cprogramming.com](https://www.cprogramming.com)
6. Learn C++. URL : <https://www.learncpp.com/>
7. C++ Programming Language. URL : <https://www.geeksforgeeks.org/c-plus-plus/>
8. C++. URL : <https://www.codingninjas.com/blog/?s=C%2B%2B>
9. C++. URL : [https://www.bestprog.net/uk/sitemap\\_ua/c/](https://www.bestprog.net/uk/sitemap_ua/c/)
10. Cplusplus. URL : <https://cplusplus.com/>