

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ



Декан

факультету інформаційних технологій

Тетяна ГОВОРУЩЕНКО

2024 р.

## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«Програмування»

**Галузь знань** – 17 Електроніка, автоматизація та електронні комунікації

**Спеціальність** – 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

**Рівень вищої освіти** – Перший бакалаврський

**Освітньо-професійна програма** – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

**Обсяг дисципліни** – 11 кредитів ЄКТС, **Шифр дисципліни** – ОПП.01

**Мова навчання** – українська


**Статус дисципліни:** обов'язкова (професійної підготовки)

**Факультет** – Інформаційних технологій

**Кафедра** – Автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

Форма здобуття освіти	Курс	Семестр	Обсяг дисципліни		Кількість годин						Курсовий проєкт	Курсова робота	Форма семестрового контролю	
			Кредити ЄКТС	Години	Аудиторні заняття					Самостійна робота, у т.ч. ІРС			Залік	Іспит
					Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття					
Д	1	1	5	150	102	34	51	17		48				+
Д	1	2	4	120	54		36	18		66			+	
Д	1	2	2	60						60	+			
<b>Разом ДФН</b>			<b>11</b>	<b>330</b>	<b>156</b>	<b>34</b>	<b>87</b>	<b>35</b>		<b>174</b>				

Робоча програма складена на основі освітньо-професійної програми «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» за спеціальністю 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

Робоча програма складена  канд. техн. наук, доцент Галина РАДЕЛЬЧУК

Схвалена на засіданні кафедри Автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

Протокол від 30 серпня 2024 р. № 1. Зав. кафедри  Валерій МАРТИНЮК

Робоча програма розглянута та схвалена вченою радою факультету інформаційних технологій

Голова вченої ради факультету  Тетяна ГОВОРУЩЕНКО

Хмельницький 2024

## ПРОГРАМУВАННЯ

Тип (статус) дисципліни	Обов'язкова професійної підготовки
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	Перший-другий
Обсяг кредитів ЄКТС	11
Форми здобуття освіти, для яких викладається дисципліна	Денна

**Результати навчання.** Після вивчення дисципліни студент повинен: *вміло використовувати* понятійний апарат; *володіти* сучасними методами і мовами програмування для розроблення алгоритмічного та програмного забезпечення (ПЗ); *аналізувати* вхідні дані завдання; *визначати* функціональні вимоги до розроблюваної програми; *аналізувати* і *порівнювати* методи розв'язання задачі та *обґрунтувати* обраний спосіб; *виконувати* алгоритмізацію задач та *розробляти* ПЗ мовами C/C++ з урахуванням особливостей базових алгоритмічних структур; *зводити* рішення задачі до вирішення підзадач; *демонструвати* культуру мислення при розробленні алгоритмів підзадач та завдання в цілому; *узагальнювати* дані при написанні коду програми; *інтерпретувати* результати роботи програми; *підбирати* тестові набори даних; *виявляти* та *усувати* програмні помилки; *оцінювати* ступінь відповідності розробленої програми визначеним вимогам.

**Зміст навчальної дисципліни.** Загальний процес комп'ютерного розв'язування задач. Базові алгоритмічні структури. Синтаксис та основні конструкції мови програмування C. Структурне програмування. Структуровані типи даних у мові C. Масиви та рядки. Вказівники. Динамічний розподіл пам'яті. Похідні типи даних. Структури та об'єднання. Процедурне програмування. Функції. Файлове введення/виведення. Модульна організація програм. Візуальне програмування мовою C++.

**Пререквізити** – вища математика; англійська мова.

**Кореквізити** – об'єктно-орієнтоване програмування; програмування мікропроцесорних систем керування; людино-машинний інтерфейс та програмування систем реального часу; технічні засоби автоматизації та робототехнічні системи; основи комп'ютерно-інтегрованих технологій та робототехніки.

**Запланована навчальна діяльність:** лекції – 34 год., лабораторні заняття – 87 год., практичні заняття – 35 год., курсове проєктування – 60 год., самостійна робота – 114 год., разом – 330 год.

**Форми (методи) навчання:** лекції (з використанням методів проблемного навчання і візуалізації); лабораторні та практичні заняття (з використанням методів інформаційних технологій та сучасних інтегрованих середовищ програмування, майстер-класів, практикумів), самостійна робота (індивідуальні практичні завдання; курсове проєктування).

**Форми оцінювання результатів навчання:** захист лабораторних робіт; презентація результатів виконання практичних завдань; тематичне онлайн-тестування; захист курсового проєкту.

**Вид семестрового контролю:** іспит – 1 семестр, залік – 2 семестр, курсовий проєкт – 2 семестр.

### Навчальні ресурси:

1. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування. Частина 1: навч. посіб. Львів : Вид-тво «Новий світ-2000», 2020. 337 с.
2. C++. Алгоритмізація та програмування: підручник. 2-ге вид. перероб. і доповн. / Трофименко О. Г. та ін. Одеса : Фенікс, 2019. 477 с.
3. Козак Л. І., Костюк І. В., Стасевич С. П. Основи програмування : навч. посіб. Львів : «Новий Світ-2000», 2020. 328 с.
4. Кривцова О. П. Програмування мовою C++. Технологія візуального програмування: навч. посіб. Полтава : ПНПУ імені В. Г. Короленка, 2020. 144 с.
5. Модульне середовище. URL: <https://msn.khmnmu.edu.ua/>
6. Електронна бібліотека. URL: [http://lib.khmnmu.edu.ua/asp/php\\_f/plage\\_lib.php](http://lib.khmnmu.edu.ua/asp/php_f/plage_lib.php)

**Викладачі:** канд. техн. наук, доц. Радельчук Галина Іванівна  
канд. техн. наук, доц. Макаришкін Денис Анатолійович

### 3 ПОЯСНЮВАЛЬНА ЗАПИСКА

Дисципліна «Програмування» є однією з фахових дисциплін і займає провідне місце у підготовці здобувачів першого (бакалаврського) рівня вищої освіти, які навчаються за освітньо-професійними програмами у межах спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

**Пререквізити** – вища математика; англійська мова.

**Кореквізити** – об'єктно-орієнтоване програмування; програмування мікропроцесорних систем керування; людино-машинний інтерфейс та програмування систем реального часу; технічні засоби автоматизації та робототехнічні системи; основи комп'ютерно-інтегрованих технологій та робототехніки.

Відповідно до *Стандарту вищої освіти* із зазначеної спеціальності та освітньої програми дисципліна сприяє розширенню і поглибленню:

**компетентностей:** здатність розв'язувати складні спеціалізовані задачі та практичні проблеми, що характеризуються комплексністю та невизначеністю умов, під час професійної діяльності у галузі автоматизації або у процесі навчання, що передбачає застосування теорій та методів галузі (інтегральна компетентність); здатність застосовувати знання у практичних ситуаціях; навички використання інформаційних і комунікаційних технологій; здатність використовувати для вирішення професійних завдань новітні технології у галузі автоматизації та комп'ютерно-інтегрованих технологій, зокрема, проєктування багаторівневих систем керування, збору даних та їх архівування для формування бази даних параметрів процесу та їх візуалізації за допомогою засобів людино-машинного інтерфейсу; здатність обґрунтовувати вибір технічної структури та вміти розробляти прикладне програмне забезпечення для мікропроцесорних систем керування на базі локальних засобів автоматизації, промислових логічних контролерів та програмованих логічних матриць і сигнальних процесорів; здатність вільно користуватись сучасними комп'ютерними та інформаційними технологіями для вирішення професійних завдань, програмувати та використовувати прикладні та спеціалізовані комп'ютерно-інтегровані середовища для вирішення задач автоматизації; здатність інтегрувати новітні технології, сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення систем автоматизації, комп'ютерно-інтегрованих технологій та робототехнічних систем.

**програмних результатів навчання:** вміти застосовувати сучасні інформаційні технології та мати навички розробляти алгоритми і комп'ютерні програми з використанням мов високого рівня та технологій об'єктно-орієнтованого програмування, створювати бази даних та використовувати інтернет-ресурси; вміти проєктувати багаторівневі системи керування і збору даних для формування бази параметрів процесу та їх візуалізації за допомогою засобів людино-машинного інтерфейсу, використовуючи новітні комп'ютерно-інтегровані технології; вміти обґрунтовувати вибір структури та розробляти прикладне програмне забезпечення для мікропроцесорних систем управління на базі локальних засобів автоматизації, промислових логічних контролерів та програмованих логічних матриць і сигнальних процесорів; використовувати інтеграцію новітніх технологій, методів створення Інтернет-ресурсів та програмування для розроблення алгоритмічного та програмного забезпечення при розв'язуванні задач проєктування і використання систем автоматизації, комп'ютерно-інтегрованих технологій та робототехнічних систем.

**Мета дисципліни.** Формування у здобувачів вищої освіти базових теоретичних знань та практичних навичок, спрямованих на вирішення завдань проєктування та розроблення ПЗ з використанням мов високого рівня.

**Предмет дисципліни.** Теорія і практика застосування базових алгоритмічних структур, базових і похідних структур даних та засобів мов програмування високого рівня при розробленні ПЗ.

**Завдання дисципліни.** Надати студентам знання і практичні навички з основ проєктування та розроблення ПЗ; сформувані базові компетентності (загальні і фахові), необхідні для подальшого вивчення циклу професійно-орієнтованих дисциплін.

**Результати навчання.** Після вивчення дисципліни студент повинен: *вміло використовувати* понятійний апарат; *володіти* сучасними методами і мовами програмування для розроблення алгоритмічного та програмного забезпечення; *аналізувати* вхідні дані завдання; *визначати* функціональні вимоги до розроблюваної програми; *аналізувати* і *порівнювати* методи розв'язання задачі та *обґрунтувати* обраний спосіб; *виконувати* алгоритмізацію задач та *розробляти* ПЗ мовами C/C++ з урахуванням особливостей базових алгоритмічних структур; *зводити* рішення задачі до вирішення підзадач; *демонструвати* культуру мислення при розробленні алгоритмів підзадач та завдання в цілому; *узагальнювати* дані при написанні коду програми; *інтерпретувати* результати роботи програми; *підбирати* тестові набори даних; *виявляти* та *усувати* програмні помилки; *оцінювати* ступінь відповідності розробленої програми визначеним вимогам.

#### 4 СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва розділу (теми)	Кількість годин, відведених на:			
	лекції	лабораторні роботи	практичні заняття	СРС
<i>Перший семестр</i>				
Тема 1. Загальний процес комп'ютерного розв'язування задач. Огляд мови C та середовища програмування Microsoft Visual Studio	6	12	2	8
Тема 2. Структурне програмування.	4	18	6	5
Тема 3. Структуровані типи даних мови C. Масиви та рядки.	6	12	6	10
Тема 4. Вказівники. Динамічний розподіл пам'яті у мові C.	4	10/8*	4/2*	5
Тема 5. Похідні типи даних мови C. Структури та об'єднання.	2			2
Тема 6. Процедурне та модульне програмування.	12			18
<b>Разом за 1-й семестр:</b>	<b>34</b>	<b>52/50*</b>	<b>18/16*</b>	<b>48</b>
<i>Другий семестр</i>				
Тема 5. Похідні типи даних мови C. Структури та об'єднання (продовження)		4	2	7
Тема 6. Процедурне та модульне програмування (продовження)		12	6	23
Тема 7. Візуальне програмування мовою C++		20	10	36
<b>Разом за 2-й семестр:</b>		<b>36</b>	<b>18</b>	<b>66</b>

*Примітка.* \* по чисельнику / по знаменнику (розрахунок здійснюється відповідно до розкладу занять)

## 5 ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### 5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
1	2	3
<i>Перший семестр</i>		
1	<p><b>Загальний процес комп'ютерного розв'язування задач. Алгоритмізація задач.</b></p> <p>Етапи розв'язування задач на ЕОМ. Алгоритм, його властивості та способи представлення. Базові алгоритмічні структури. Поняття програми, мови програмування, транслятора, системи програмування. Тестування та відладка програми.</p> <p>Літ.: [1] с. 18-41; [2] с. 8-14; [3] с. 117-134</p>	2
2	<p><b>Мова програмування C/C++. Основні конструкції. Структура програми. Середовище розроблення програм.</b></p> <p>Загальний огляд C/C++. Синтаксис C. Типи даних. Змінні та константи. Стандартні математичні функції. Операції та вирази. Перетворення типів. Структура C-програми. Середовище розроблення Microsoft Visual Studio.</p> <p>Літ.: [1] с. 45-58; [2] с. 15-64; [3] с. 11-116</p>	2
3	<p><b>Консольне введення/виведення даних.</b></p> <p>Організація консольного введення та виведення даних в C/C++. Форматоване введення/виведення даних. Функції printf() та scanf(). Введення/виведення символів та рядків. Введення/виведення у стилі C++. Локалізація консольних додатків.</p> <p>Літ.: [1] с. 77-91</p>	2
4	<p><b>Розроблення консольних додатків лінійної та розгалуженої структури</b></p> <p>Основні правила конструювання лінійної програми. Оператори слідування. Складений оператор (блок). Прийняття рішень (розгалуження). Логічні вирази. Оператори, призначені для організації розгалужень (if, if...else, goto, switch, break). Типові приклади.</p> <p>Літ.: [1] с. 66-76, 103-110; [2] с. 65-101; [3] с. 135-158</p>	2
5	<p><b>Програмування циклічних алгоритмічних структур</b></p> <p>Арифметичні та ітераційні цикли. Оператори циклу for, while, do...while. Вкладені цикли. Оператори break та continue. Типові приклади.</p> <p>Літ.: [1] с. 124-134; [2] с. 102-148; [3] с. 159-170</p>	2
6	<p><b>Числові масиви в C. Робота з одновимірними масивами</b></p> <p>Поняття масиву у мові C. Одновимірні масиви. Оголошення та ініціалізація. Формування, опрацювання та виведення одновимірних масивів. Наслідки виходу за межі діапазону зміни індекса. Типові приклади.</p> <p>Літ.: [1] с. 217-220; [2] с. 149-175; [3] с. 195-214</p>	2
7	<p><b>Числові масиви в C. Робота з багатовимірними масивами</b></p> <p>Організація двовимірних та багатовимірних масивів. Оголошення та ініціалізація. Опрацювання двовимірних масивів. Опрацювання багатовимірних масивів. Типові приклади.</p> <p>Літ.: [1] с. 221-232; [2] с. 175-205</p>	2
8	<p><b>Символьні масиви в C (рядки). Масиви рядків</b></p> <p>Робота з текстом. Деякі прийоми опрацювання символьних даних. Символьні масиви (рядки). Оголошення та ініціалізація. Введення, опрацювання та виведення символьних масивів. Бібліотечні функції мови C для роботи з рядками. Масиви рядків. Типові приклади.</p> <p>Літ.: [1] с. 276-289; [2] с. 240-304; [3] с. 287-294</p>	2

1	2	3
9	<b>Вказівники. Вказівники та масиви</b> Змінні, адреси та вказівники. Оголошення та ініціалізація вказівників. Операції над вказівниками. Вказівники та масиви. Вказівники та рядки. Масиви вказівників. Багаторівнева адресація. Типові приклади. Літ.: [1] с. 144-149, 225-232; [2] с. 206-215; [3] с. 245-268	2
10	<b>Динамічний розподіл пам'яті. Функції динамічного розподілу пам'яті та їх застосування</b> Способи виділення пам'яті для об'єктів у мові С. Динамічна пам'ять. Функції динамічного розподілу пам'яті. Моделювання одновимірних масивів з використанням динамічної пам'яті. Динамічне виділення пам'яті для багатовимірних масивів. Опрацювання динамічних масивів. Літ.: [1] с. 250-263; [2] с. 216-239	2
11	<b>Похідні типи даних мови С. Структури та об'єднання</b> Способи створення похідних типів даних у мові С. Тип перерахування. Переіменовування типів (typedef). Структури (struct). Масиви структур. Вкладені структури. Об'єднання (union). Бітові поля. Літ.: [1] с. 297-305; [2] с. 362-379	2
12	<b>Функції у мові С. Оголошення (опис) та визначення функції. Обмін даними між функціями</b> Поняття функції. Оголошення та визначення функції. Оператор return. Параметри функцій: формальні, фактичні параметри, локальні та глобальні змінні. Виклик функції. Способи передачі даних у функцію (за значенням, за адресою). Функції зі змінним числом параметрів. Літ.: [1] с. 153-167; [2] с. 305-321; [3] с. 215-244	2
13	<b>Функції і вказівники</b> Масиви як параметри функцій. Програмування функцій, аргументами яких можуть бути вказівники. Функції, які повертають значення через вказівник. Вказівники на функції. Літ.: [1] с. 225-232; [3] с. 215-244	2
14	<b>Опрацювання структур у функціях</b> Способи передачі структур у функції, Створення функцій, які повертають структури і вказівники на структури. Літ.: [1] с. 302; [3] с. 215-244	2
15	<b>Рекурсивні алгоритми та функції</b> Поняття рекурсії. Види рекурсії та застосування рекурсивних алгоритмів. Програмування мовою С з використанням рекурсивних функцій. Літ.: [1] с. 180-188; [2] с. 322-336	2
16	<b>Файлове введення/виведення</b> Поняття потоку і файлу в С. Текстові та бінарні файли. Файлова система ANSI C. Функції файлового введення/виведення. Режими відкриття файлів. Перевірка на помилки відкриття файлів. Закриття файла. Запис/читання інформації. Форматоване введення/виведення. Введення/виведення по блоках. Прямий (довільний) доступ до файлів. Літ.: [1] с. 314-327; [2] с. 402-455	2
17	<b>Модульна організація програм</b> Поняття модуля і модульного програмування. Структура складної С-програми. Багатофайлові програми. Міжфайлова взаємодія. Заголовні файли (.h-файли) і файли реалізації (.c-файли). Директиви препроцесора. Стандартні бібліотеки (.lib). Літ.: [1] с. 201-208; [2] с. 337-361	2
<b>Разом:</b>		<b>34</b>

## 5.2 Зміст лабораторних (практичних, семінарських) занять

### Перелік лабораторних занять для студентів денної форми здобуття освіти

№ з/п	Тема лабораторного заняття	Кількість годин
<i>1</i>	<i>2</i>	<i>3</i>
<i>Перший семестр</i>		
1	Алгоритмізація задач Літ.: [6]	6
2	Вивчення основних прийомів роботи у середовищі Microsoft Visual Studio Літ.: [6]	6
3	Розроблення С-програм лінійної структури Літ.: [5] с. 5-27; [6]	6
4	Розроблення С-програм розгалуженої структури Літ.: [5] с. 33-68; [6]	6
5	Розроблення С-програм циклічної структури Літ.: [5] с. 72-102; [6]	6
6	Опрацювання одновимірних масивів та рядків Літ.: [5] с. 122-140, 201-228; [6]	6
7	Робота з багатовимірними масивами Літ.: [5] с. 141-163; [6]	6
8	Робота з вказівниками. Опрацювання динамічних масивів Літ.: [5] с. 164-184; [6]	6
9	Підсумкове заняття	4/2*
<b>Разом:</b>		<b>52/50*</b>
<i>Другий семестр</i>		
1	Опрацювання структур Літ.: [5] с. 252-260; [6]	4
2	Розроблення С-програм з використанням функцій користувача Літ.: [5] с. 105-121, 185-200; [6]	8
3	Робота з файлами даних Літ.: [5] с. 229-250, 276-298; [6]	4
4	Створення проєкту застосунку Windows Forms засобами мови С++. Програмування елементів керування. Літ.: [4] с. 7-30, 88-102	4
5	Розроблення графічного інтерфейсу користувача Літ.: [4] с. 31-38, 80-82	8
6	Розроблення програмних застосунків з використанням графіки Літ.: [4] с. 39-69, 103-113, 128-135	8
<b>Разом:</b>		<b>36</b>

**Примітка.** \* по чисельнику / по знаменнику (розрахунок здійснюється відповідно до розкладу занять)

**Перелік практичних занять для студентів денної форми здобуття освіти**

№ з/п	Тема практичного заняття	Кількість годин
1	2	3
<b>Перший семестр</b>		
1	Вивчення та розроблення базових алгоритмічних структур. Розбиття задачі на підзадачі. Літ.: [1] с. 18-41; [2] с. 8-14	2
2	Основні конструкції мови С. Програмування алгебраїчних виразів та використання математичних функцій бібліотеки мови С. Програмування лінійних алгоритмів. Типові задачі. Літ.: [1] с. 66-76; [2] с. 65-70; [3] с. 126-134	2
3	Програмування розгалужених алгоритмів. Типові задачі. Літ.: [1] с. 103-110; [2] с. 71-101; [3] с. 135-158	2
4	Оператори циклу у мові С. Отримання практичних навиків у роботі з операторами циклу. Програмування з використанням вкладених циклів [1] с. 124-134; [2] с. 102-148; [3] с. 159-170	2
5	Одновимірні масиви. Класи типових задач. Літ.: [1] с. 217-220; [2] с. 149-175; [3] с. 195-214	2
6	Багатовимірні масиви. Робота з матрицями. Літ.: [1] с. 221-232; [2] с. 175-205	2
7	Робота з символами та символічними масивами (рядками) Літ.: [1] с. 276-289; [2] с. 240-304; [3] с. 287-294	2
8	Робота з вказівниками. Динамічні масиви. Літ.: [1] с. 144-263, 225-232; [2] с. 206-239; [3] с. 245-268	2
9	Підсумкове заняття	2/0*
<b>Разом:</b>		<b>18/16*</b>
<b>Другий семестр</b>		
1	Опрацювання структур та об'єднань Літ.: [1] с. 297-305; [2] с. 362-379	2
2	Функції користувача. Базові прийоми роботи. Літ.: [1] с. 153-167; [2] с. 305-321; [3] с. 215-244	2
3	Функції і вказівники. Опрацювання структурних типів даних у функціях. Рекурсія. Літ.: [1] с. 180-188, 225-232; [2] с. 322-336; [3] с. 215-244	2
4	Файлове введення/виведення Літ.: [1] с. 314-327; [2] с. 402-455	2
5	Створення проєкту застосунку Windows Forms засобами мови С++. Інструменти візуального розроблення застосунків. Літ.: [4] с. 7-30, 76-79	2
6	Створення головного меню, контекстного меню та спливаючих підказок. Створення багатовіконних застосунків. Літ.: [4] с. 7-38, 91-102	2
7	Графічні можливості мови С++. Літ.: [4] с. 39-69	4
8	Опрацювання виключень Літ.: [3] с. 305-333; [7] розділ 14	2
<b>Разом:</b>		<b>18</b>

**Примітка.** \* по чисельнику / по знаменнику (розрахунок здійснюється відповідно до розкладу занять)



У процесі виконання лабораторних та практичних робіт з дисципліни студенти набувають практичних навичок із: алгоритмізації задач; розроблення прикладного ПЗ мовою високого рівня та його налагодження; користування спеціальними інструментальними засобами та інтегрованими середовищами програмування.

### 5.3 Зміст самостійної (у т. ч. індивідуальної) роботи

Самостійна робота студентів полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації, підготовці до практичних занять, виконанні домашніх практичних завдань, підготовці до виконання та захисту лабораторних робіт, підготовці до тестування з теоретичного та практичного матеріалу тощо.

Керівництво самостійною роботою студентів здійснюється викладачем згідно з розкладом консультацій у позаурочний час.

#### Зміст самостійної роботи студентів денної форми здобуття освіти

Номер тижня	Вид самостійної роботи	Кількість годин
<i>Перший семестр</i>		
<i>1</i>	<i>2</i>	<i>3</i>
1	Опрацювання теоретичного матеріалу. Підготовка до виконання ЛР1. Підготовка до ПЗ1.	2
2	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР1. Підготовка до виконання ЛР2. Підготовка до ПЗ2	3
3	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР2.	3
4	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР2. Підготовка до виконання ЛР3. Підготовка до ПЗ3.	2
5	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР3. Підготовка до виконання ЛР4.	3
6	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР3. Підготовка до виконання ЛР4. Підготовка до ПЗ4.	2
7	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР4. Підготовка до виконання ЛР5. Підготовка до тестового контролю з тем 1-2.	4
8	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР4. Підготовка до виконання ЛР5. Підготовка до тестового контролю з тем 1-2. Підготовка до ПЗ5.	4
9	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР5. Підготовка до виконання ЛР6.	2
10	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР5. Підготовка до виконання ЛР6. Підготовка до ПЗ6.	3
11	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР6. Підготовка до виконання ЛР7.	2
12	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР6. Підготовка до виконання ЛР7. Підготовка до ПЗ7.	3
13	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР7. Підготовка до виконання ЛР8.	2
14	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР7. Підготовка до виконання ЛР8. Підготовка до ПЗ8.	3
15	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР8. Підготовка до тестового контролю з тем 3-4.	4

1	2	3
16	Опрацювання теоретичного матеріалу. Підготовка до тестового контролю з тем 3-4.	4
17	Опрацювання теоретичного матеріалу.	2
<b>Разом:</b>		<b>48</b>
<i>Другий семестр</i>		
1	Опрацювання теоретичного матеріалу. Підготовка до виконання ЛР9. Підготовка до ПЗ9.	3
2	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР9. Підготовка до виконання ЛР10.	4
3	Опрацювання теоретичного матеріалу. Підготовка до виконання ЛР10. Підготовка до ПЗ10.	3
4	Опрацювання теоретичного матеріалу.	3
5	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР10. Підготовка до ПЗ11.	4
6	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР10. Підготовка до виконання ЛР11.	4
7	Опрацювання теоретичного матеріалу. Підготовка до тестового контролю з тем 5-6. Підготовка до ПЗ12.	5
8	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР11. Підготовка до виконання ЛР12. Підготовка до тестового контролю з тем 5-6.	4
9	Опрацювання теоретичного матеріалу. Підготовка до ПЗ13.	3
10	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР12. Підготовка до виконання ЛР13.	4
11	Опрацювання теоретичного матеріалу. Підготовка до ПЗ14.	3
12	Опрацювання теоретичного матеріалу.	3
13	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР13. Підготовка до ПЗ15.	4
14	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР13. Підготовка до виконання ЛР14.	4
15	Опрацювання теоретичного матеріалу. Підготовка до тестового контролю з теми 7. Підготовка до ПЗ15.	5
16	Опрацювання теоретичного матеріалу. Підготовка до тестового контролю з теми 7.	3
17	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР14. Підготовка до ПЗ16.	4
18	Опрацювання теоретичного матеріалу.	3
<b>Разом:</b>		<b>66</b>

*Скорочення:* ЛР – лабораторна робота; ПЗ – практичне заняття.

## 6 ТЕХНОЛОГІЇ ТА МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій, зокрема: лекції (з використанням методів проблемного навчання і візуалізації); лабораторні та практичні заняття (з використанням методів інформаційних технологій та сучасних інтегрованих середовищ програмування, майстер-класів, практикумів), самостійна робота

(індивідуальні практичні завдання, курсове проєктування), і мають за мету – оволодіння студентами спеціальною термінологією; набуття ними знань та практичних навичок з алгоритмізації задач та програмування мовою високого рівня; формування базових компетентностей, необхідних для подальшого вивчення циклу професійно-орієнтованих дисциплін.

Необхідні інструменти, обладнання, програмне забезпечення: комп'ютерна техніка; офісне ПЗ; інтегроване середовище програмування Microsoft Visual Studio.

## 7 МЕТОДИ КОНТРОЛЮ

Поточний контроль здійснюється під час практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою та графіком навчального процесу. При цьому використовуються такі методи поточного контролю: усне опитування перед допуском до лабораторного заняття; захист лабораторних робіт; тематичний тестовий контроль; презентація виконаних домашніх практичних завдань.

При виведенні підсумкової семестрової оцінки враховуються результати як поточного контролю, так і підсумкового контрольного заходу, який проводиться як письмова підсумкова робота з усього матеріалу дисципліни. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід, вважається невстигаючим.

## 8 ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ СТУДЕНТІВ У СЕМЕСТРІ

Оцінювання академічних досягнень здобувача вищої освіти здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». Кожний вид роботи з дисципліни оцінюється за інституційною **чотирибальною** шкалою («відмінно», «добре», «задовільно», «незадовільно») і виставляється в електронному журналі обліку успішності. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих студентом **позитивно**, з урахуванням коефіцієнта вагомості і розраховується в автоматизованому режимі за відповідною програмою. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих видів навчальної роботи для формування компетентностей і забезпечення програмних результатів навчання.

Оцінка, яка виставляється за лабораторну роботу, складається з таких елементів: усне опитування студентів перед допуском до виконання лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення звіту з лабораторної роботи; вільне володіння студентом спеціальною термінологією і уміння професійно обґрунтувати прийняті рішення; своєчасний захист лабораторної роботи.

Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. Пропущене лабораторне заняття студент зобов'язаний відпрацювати у лабораторіях кафедри у встановлений викладачем термін з реєстрацією у відповідному журналі кафедри, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Засвоєння студентом теоретичного та практичного матеріалу з дисципліни оцінюється тестуванням.

Оцінювання знань студентів здійснюється за такими критеріями:

<i>Оцінка за інституційною шкалою</i>	<i>Узагальнений критерій</i>
<b>Відмінно</b>	Студент глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і вміло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає грамотний, логічний виклад відповіді державною мовою (як в усній, так і у письмовій формі), якісне зовнішнє оформлення роботи. Студент не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки. При відповіді допустив дві-три несуттєві похибки.

<b>Добре</b>	Студент виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних задач; виклад відповіді грамотний, але у змісті відповіді можуть мати місце окремі неточності, нечіткі формулювання закономірностей тощо. Відповідь студента будується на основі самостійного мислення. Студент у відповіді допустив дві-три несуттєві помилки.
<b>Задовільно</b>	Студент виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент має слабкі знання структури курсу, допускає неточності і суттєві помилки у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.
<b>Незадовільно</b>	Студент виявив розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускає помилки у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка «незадовільно» виставляється студенту, який не може продовжити навчання без додаткової роботи з вивчення дисципліни.

### Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми здобуття освіти у семестрі за ваговими коефіцієнтами

Аудиторна робота	Самостійна, індивідуальна робота		Семестровий контроль, іспит
<i>Перший семестр</i>			
Лабораторні роботи	Тестовий контроль		Підсумковий контрольний захід
№ 1 – 8	TK1 (T1-2)	TK2 (T3-4)	1
<b>ВК: 0,4</b>	<b>0,2</b>		<b>0,4</b>
<i>Другий семестр</i>			
Лабораторні роботи	Тестовий контроль		Семестровий контроль, залік
№ 9 – 14	TK3 (T5-6)	TK4 (T7)	За рейтингом
<b>0,8</b>	<b>0,2</b>		<b>0</b>

Умовні позначення: ВК – ваговий коефіцієнт, ТК – тестовий контроль, Т – тема дисципліни.

### Оцінювання тестових завдань

Тематичний тест для кожного студента складається з 25 тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 25.

Оцінювання здійснюється за чотирибальною шкалою.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту:

Сума балів за тестові завдання	1 – 13	14 – 16	17 – 22	23 – 25
Оцінка за 4-бальною шкалою	2	3	4	5

Тестування студент проходить в онлайн-режимі у модульному середовищі для навчання Moodle. На тестування відводиться 30 хвилин.

При отриманні негативної оцінки тест слід перездати до терміну наступного контролю.

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення викладачем усіх оцінок до електронного журналу.

Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені у таблиці.

Залік виставляється, якщо середньозважений бал, який отримав студент з дисципліни, знаходиться у межах від 3,00 до 5,00 балів. При цьому за інституційною шкалою ставиться оцінка «зараховано», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом кількості балів відповідно до таблиці Співвідношення.

### Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Інституційна інтервальна шкала балів	Інституційна оцінка, критерії оцінювання		
A	4,75–5,00	5	Зараховано	<b>Відмінно</b> – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навичок
B	4,25–4,74	4		<b>Добре</b> – повне знання навчального матеріалу з кількома незначними помилками
C	3,75–4,24	4		<b>Добре</b> – в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25–3,74	3		<b>Задовільно</b> – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00–3,24	3		<b>Задовільно</b> – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00–2,99	2	Незараховано	<b>Незадовільно</b> – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00–1,99	2		<b>Незадовільно</b> – необхідна серйозна подальша робота і повторне вивчення дисципліни

### Курсове проєктування

Індивідуальна робота студентів у другому семестрі відповідно до навчального плану передбачає виконання курсового проєкту (КП).

Метою курсового проєктування є поглиблення та систематизація теоретичних знань і практичних навичок з алгоритмізації задач та програмування, набутих студентом при вивченні дисципліни «Програмування», застосування їх для комплексного вирішення практичних задач.

Тематика КП щорічно розробляється керівником і затверджується на засіданні кафедри.

Завдання видається студенту на останньому тижні першого семестру, а виконання та захист КП – у другому семестрі відповідно до графіка. КП студенти виконують згідно із методичними рекомендаціями до курсового проєктування.

У процесі роботи над КП студент має проявити творчу ініціативу, креативне мислення та самостійність при обґрунтуванні та виборі методів/алгоритмів розв'язування задач, використовуючи при цьому знання та практичні навички зі спеціальних та загальноінженерних дисциплін.

Обсяг розрахунково-пояснювальної записки (з додатками) – до 30–35 сторінок машинописного тексту. Обсяг самостійної роботи студента – 60 год. Консультації з КП здійснює керівник проєкту – викладач кафедри – відповідно до графіка, затвердженого деканом факультету. Захист

КП здійснюється (після його перевірки керівником) перед комісією, яка складається з 2–3 викладачів кафедри. До складу комісії обов'язково входить керівник проєкту.

### Узагальнена орієнтовна тематика курсових проєктів

1. Розроблення діалогової або тестової програми.
2. Розроблення програми введення та опрацювання файлу з друком вихідної форми.
3. Розроблення навчальних тестових програм з інтерфейсами сучасних систем.
4. Розроблення програм для вирішення прикладних задач.

### Оцінювання курсового проєкту

Оцінювання КП здійснюється за інституційною чотирибальною шкалою та шкалою ЄКТС за видами робіт і ваговими коефіцієнтами. Оцінка «відмінно/А» виставляється за високоякісно виконані пояснювальну записку і програму, де немає помилок, за оригінальність рішень, дотримання вимог методичних рекомендацій з курсового проєктування. Доповідь і захист проєкту обґрунтовані, виявлені комплексні знання з дисципліни стосовно теми КП.

Оцінка «добре/В» виставляється за якісне виконання КП при одній-двох незначних помилках чи не досить впевнені відповіді на одне-два питання комісії.

Оцінка «добре/С» виставляється за якісно виконаний проєкт, дотримання усіх вимог, що пред'являються до КП, за дві-три незначні помилки у пояснювальній записці/програмі, не чіткі відповіді на два-три питання комісії.

Оцінка «задовільно/Д» виставляється, якщо у пояснювальній записці чи програмі виявлені помилки, є незначні порушення вимог до оформлення проєкту, невпевнені відповіді на основні питання з теми проєкту.

Оцінка «задовільно/Е» виставляється, якщо у проєкті виявлені суттєві помилки як у пояснювальній записці, так і в програмі, неправильно обґрунтовані прийняті алгоритмічні та програмні рішення, грубі помилки при відповідях на запитання членів комісії, невпевненому захисті КП у цілому.

Оцінка «незадовільно/ФХ/Ф» виставляється, якщо в проєкті вибрані неправильні методи (алгоритми) вирішення задачі, або за невідповідність змісту і програми затвердженій темі КП, коли студент не орієнтується в тому, що виконав. У цьому випадку студент представляє виправлений проєкт на повторний захист, або йому видається нова тема проєкту і призначається термін його виконання та захисту.

### Структурування курсового проєкту за видами робіт і ваговими коефіцієнтами

Якість виконання КП		Якість захисту КП	
Пояснювальна записка	Програма	Доповідь, презентація	Відповіді на запитання
ВК: 0,4	0,3	0,2	0,1

## 9 ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ РЕЗУЛЬТАТІВ НАВЧАННЯ

1. Основні етапи розв'язування задач за допомогою ПК.
2. Базові алгоритмічні структури.
3. Транслятор. Види трансляторів, їх принципи роботи, достоїнства та недоліки.
4. Алфавіт та лексеми мови C.
5. Стандартні типи даних мови C.
6. Змінні та константи мови C. Оголошення та ініціалізація.
7. Стандартні математичні функції мови C.
8. Вирази та арифметичні операції мови C. Чим відрізняються префіксне і постфіксне інкрементування і декрементування?
9. Перетворення типів у виразах. Неявне і явне приведення типів.
10. Загальна структура C-програми. Яке значення повертає головна функція `main()` проекту у C-програмах?
11. Загальна схема підготовки виконуваної програми (.exe-файла). Які основні засоби для цього необхідні?
12. Інтегроване середовище розробки Microsoft Visual Studio. Рішення і проект. У якому вікні Microsoft Visual Studio міститься графічне представлення всього рішення?
13. Форматоване введення даних. Функція `scanf()`. Основні специфікатори формату.
14. Форматоване виведення даних. Функція `printf()`. Основні специфікатори формату.
15. Введення/виведення символів. Функції `getchar()` і `putchar()`.
16. Введення/виведення рядків. Функції `gets()` і `puts()`.
17. Локалізація консольних додатків у мові C.
18. Охарактеризуйте оператори слідування в лінійній програмі.
19. Логічні операції та вирази.
20. Охарактеризуйте умовний оператор `if`. Для чого призначений оператор `goto`?
21. Охарактеризуйте оператор вибору `switch`.
22. Охарактеризуйте тернарний оператор `?:`.
23. Синтаксис та порядок виконання циклу `for`.
24. Синтаксис та порядок виконання циклу `while`. Яку мінімальну кількість разів може виконатись цикл `while`?
25. Синтаксис та порядок виконання циклу `do...while`. Яку мінімальну кількість разів може виконатись цикл `do...while`?
26. Дострокове завершення циклу. Оператори `break` та `continue`. У чому відмінність і подібність між цими операторами?
27. Одновимірні масиви у мові C. Оголошення та ініціалізація.
28. Формування, опрацювання та виведення одновимірних масивів. На кого чи на що покладається контроль границь числових масивів?
29. Двовимірні масиви у мові C. Оголошення та ініціалізація. Яка мінімальна кількість циклів потрібна для роботи з двовимірними масивами?
30. Формування, опрацювання та виведення двовимірних масивів. На кого чи на що покладається контроль границь числових масивів?
31. Символьні масиви (рядки). Оголошення та ініціалізація. Що таке константа EOF і з якою метою вона використовується?
32. Введення/виведення символьних масивів (рядків). Особливості введення рядків за допомогою функції `scanf()`.
33. Основні бібліотечні функції мови C для роботи з рядками.
34. Масиви рядків. Оголошення та ініціалізація.
35. Змінні, адреси та вказівники. Оголошення та ініціалізація вказівників. Який сенс має значення вказівника NULL?
36. Основні операції над вказівниками.
37. Вказівники та одновимірні масиви.

38. Вказівники та рядки.
  39. Вказівники та багатовимірні масиви.
  40. Масиви вказівників, оголошення та ініціалізація. Багаторівнева адресація.
  41. Способи виділення пам'яті для об'єктів у мові С. Динамічний розподіл пам'яті.
  42. Функції динамічного розподілу пам'яті у мові С. Яка різниця у дії функцій **malloc()** і **calloc()**?
  43. Моделювання одновимірних масивів з використанням динамічної пам'яті.
  44. Динамічне виділення пам'яті для багатовимірних масивів (1-й спосіб).
  45. Динамічне виділення пам'яті для багатовимірних масивів (2-й спосіб).
  46. Вільні масиви. Особливості виділення динамічної пам'яті під вільний масив.
  47. Охарактеризуйте тип даних “перерахування”.
  48. Поняття структури у мові С. Оголошення та ініціалізація. У чому полягає відмінність структури від масиву?
  49. Операції зі структурами та з елементами структур.
  50. Масиви структур та елементи-масиви.
  51. Вкладені структури.
  52. Об'єднання. Оголошення та ініціалізація.
  53. Функції у мові С. Оголошення (опис) та визначення функції.
  54. Параметри функцій (глобальні, локальні, формальні та фактичні). Виклик функції.
  55. Загальні правила організації функцій у мові С.
  56. Способи передачі даних у функцію. Оператор **return**.
  57. Функції зі змінним числом параметрів.
  58. Одновимірні масиви як параметри функцій. Чи може функція за допомогою **return** повернути масив?
  59. Двовимірні масиви як параметри функцій.
  60. Функції, які повертають значення через вказівник.
  61. Опрацювання структур у функціях.
  62. Охарактеризуйте рекурсивні функції.
  63. Потоки і файли у мові С. Стандартні потоки. Текстові та бінарні файли.
  64. Файлова система ANSI C. Загальна характеристика.
  65. Вказівник на файл. Відкриття та закриття файла. Як у мові С кодується ознака кінця файла?
  66. Запис у файл та читання з файла символів. Функції **getc()** і **fgetc()**.
  67. Запис у файл та читання з файла рядків. Функції **fprintf()** і **fscanf()**.
  68. Синтаксис та призначення функцій **feof()** та **rewind()** при роботі з файлами.
  69. Читання/запис блоків даних. Функції **fread()** і **fwrite()**.
  70. Довільний (прямий) доступ до файлів і функція **fseek()**.
  71. Охарактеризуйте суть модульного програмування.
  72. Призначення препроцесора мови С. Основні директиви препроцесора.
  73. Багатофайлові програми.
  74. Специфікатори класів пам'яті при оголошенні змінних у багатофайловій програмі.
- Область видимості змінних.
75. Заголовні файли (h-файли) та їх основні елементи.

## 10 НАВЧАЛЬНО- МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Освітній процес з дисципліни «Програмування» повністю і в достатній кількості забезпечений необхідними навчально-методичними матеріалами у Модульному навчальному середовищі. Доступ до ресурсу: <https://msn.khmnu.edu/course/view.php?id=9340>.

Також підготовлена і видана методична розробка: Програмування. Лабораторний практикум / Г. І. Радельчук. Хмельницький: ХНУ, 2023. 205 с.



## 11 РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### Основна

1. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування. Частина 1: навч. посіб. Львів : Вид-тво «Новий світ-2000», 2020. 337 с.
2. С++. Алгоритмізація та програмування: підручник. 2-ге вид. перероб. і доповн. / Трофименко О. Г. та ін. Одеса : Фенікс, 2019. 477 с.
3. Козак Л. І., Костюк І. В., Стасевич С. П. Основи програмування : навч. посіб. Львів : «Новий Світ-2000», 2020. 328 с.
4. Кривцова О. П. Програмування мовою С++. Технологія візуального програмування: навч. посіб. Полтава : ПНПУ імені В. Г. Короленка, 2020. 144 с.
5. Трофименко О. Г., Прокоп Ю. В., Задерейко О. В. Алгоритмізація та програмування : навч.-метод. посіб. Одеса : Фенікс, 2020. 310 с.
6. Програмування. Лабораторний практикум / Г. І. Радельчук. Хмельницький : ХНУ, 2023. 205 с.

### Додаткова

7. Уроки програмування на С++. URL: <https://acode.com.ua/uroki-po-cpp/>
8. Мова програмування С++. URL: <http://cpp.dp.ua/>
9. Керніган Браян В., Деніс М. Річі Мова програмування С; 2-ге видання. URL: <http://programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
10. Seacord R. Effective C: An Introduction to Professional C Programming. ProgrammingPaperback. San Francisco, California, No Starch Press. 2020. 272 p.
11. C Programming Language Tutorial. URL: <https://www.geeksforgeeks.org/c-programming-language/>. Last Updated: 19 Apr, 2023.
12. Brian H. Beej's Guide to C Programming. URL: [https://beej.us/guide/bgc/pdf/bgc\\_usl\\_c\\_1.pdf](https://beej.us/guide/bgc/pdf/bgc_usl_c_1.pdf)
13. С++ Підручник. URL: <https://w3schoolsua.github.io/cpp/index.html#gsc.tab=0>
14. Прокопенко О. В., Попов М. О., Чумак Г. Л. Мова програмування С/С++. Практикум: навч. посіб. К.: Київський національний університет імені Тараса Шевченка, 2024. 375 с.
15. Стандарт С11. URL: <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1570.pdf>

## 12 ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання.  
Доступ до ресурсу: <https://msn.khmnu.edu.ua/course/view.php?id=9340>
2. Електронна бібліотека університету.  
Доступ до ресурсу: [http://lib.khmnu.edu.ua/asp/php\\_f/p1age\\_lib.php](http://lib.khmnu.edu.ua/asp/php_f/p1age_lib.php)
3. Репозитарій ХНУ. Доступ до ресурсу: <https://elar.khmnu.edu.ua/home>