

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
Кафедра інформаційних технологій

«ЗАТВЕРДЖУЮ»

Ректор Міжнародного гуманітарного

університету д.ю.н., професор

Костянтин ГРОМОВЕНКО

«29» серпня 2025 р.



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Основи програмування

Галузь знань	<u>Е Інформаційні технології</u>
Спеціальність	<u>Е5 Кібербезпека та захист інформації</u>
Назва освітньої програми	<u>Кібербезпека</u>
Рівень вищої освіти	<u>перший (бакалаврський) рівень</u>

Одеса - 2025 рік

Робоча програма затверджена на засіданні кафедри інформаційних технологій
Протокол № 1 від 28.08.2025 року.

Розробники і викладачі	Контактний тел.	E-mail
Доктор технічних наук, професор Приходько Сергій Борисович	+380993876131	sergiy.prykhodko@nuos.edu.ua

Завідувачка кафедри інформаційних
технологій к.т.н., доцент

 Тетяна ГРИГОР'ЄВА

Гарант освітньої програми
к.т.н., доцент

 Лариса ЙОНА

Узгоджено

Начальник навчального відділу

 Лілія САЄНКО

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 10 Загальна кількість годин – 300	Галузь - F Інформаційні технології Спеціальність – F5 Кібербезпека та захист інформації	обов'язкова	
		Рік підготовки:	
		1-й	
		Семестр	
		1-й, 2-й	1-й, 2-й
Мова навчання – українська	Рівень вищої освіти – перший (бакалаврський) рівень	Лекції	
		62 год.	12 год.
		Практичні, семінарські	
		62 год.	12 год.
		Лабораторні	
		-	-
		Самостійна робота та індивідуальні завдання	
		176 год.	276 год.
		Вид контролю:	
		Залік, екзамен	Залік, екзамен

Курс "Основи програмування" є необхідним практичним і теоретичним фундаментом для фахівців з комп'ютерних наук, спрямована на вивчення основ алгоритмізації, принципів опрацювання цифрової інформації, формує навички програмування алгоритмічною мовою програмування високого рівня C++. Під час вивчення даної дисципліни здобувачі здобудуть знання, які допоможуть ефективно використовувати сучасні розробки технологій програмування при вивченні спеціальних дисциплін. Вивчається одна з найпоширеніших у світі мов об'єктно-орієнтованого програмування – мова C++. На практичних заняттях створюються консольні застосунки мовою C++.

Метою викладання навчальної дисципліни є формування у здобувачів алгоритмічного мислення, практичне засвоєння основних понять і концепцій програмування та набуття навичок розробки програм мовою програмування C++ для вирішенні прикладних задач із різних предметних областей. Завданням дисципліни є набуття здобувачами теоретичних знань та практичних навичок з програмування сучасними алгоритмічними мовами високого рівня. В системі підготовки фахівця дисципліна займає особливе місце, оскільки засвоєння курсу складає перший найважливіший крок, абсолютно необхідний для успішного засвоєння подальших фахових дисциплін.

Метою викладання навчальної дисципліни є оволодіння здобувачами навиками створення широкої теоретичної підготовки в області фізики, що дозволить їм орієнтуватися в потоці наукової і технічної інформації.

ПРЕРЕКВІЗИТИ:

Знання:

- предмету і сутності комп'ютерних технологій обробки даних;
- особливостей роботи операційної системи Windows;
- основ вищої математики та дискретного аналізу;
- основ методології математичного моделювання економічних процесів;
- механізмів застосування теоретичних методів і моделей у відображенні економічних процесів.

Вміння:

- вільно працювати з файловою структурою операційної системи Windows;
- працювати з основними програмами пакету MS Office;

- розуміти математичні моделі структурованих даних.

ПОСТРЕКВІЗИТИ. Знання, уміння та навички, набуті під час вивчення дисципліни, використовуються при подальшому вивченні дисциплін бакалаврської підготовки, практики та написанні кваліфікаційної роботи.

2. ОЧІКУВАНІ КОМПЕТЕНТНОСТІ, ЯКІ ПЛАНУЄТЬСЯ СФОРМУВАТИ ТА ДОСЯГНЕННЯ ПРОГРАМНИХ РЕЗУЛЬТАТІВ

У процесі реалізації програми дисципліни «Основи програмування» формуються наступні компетентності із передбачених освітньою програмою:

Інтегральна компетентність

ІК. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі кібербезпеки та захисту інформації.

Загальні компетентності (ЗК)

ЗК1. Здатність застосовувати знання у практичних ситуаціях.

ЗК 5. Здатність вчитися і оволодівати сучасними знаннями

Спеціальні (фахові, предметні) компетентності

СК 2. Здатність використовувати інформаційні технології, сучасні методи і моделі кібербезпеки та системи захисту інформації

Навчальна дисципліна «Основи програмування» забезпечує досягнення програмних результатів навчання (РН), передбачених освітньою програмою:

РН 5. Аналізувати, аргументувати, приймати рішення при розв'язанні складних спеціалізованих задач і практичних завдань у професійній діяльності, які характеризуються комплексністю та неповною визначеністю умов, відповідати за прийняті рішення

РН 6. Адаптуватися до нових умов і технологій професійної діяльності, прогнозувати кінцевий результат.

РН 8. Застосовувати знання и розуміння математики та фізики в професійній діяльності, формалізувати задачі предметної галузі кібербезпеки та захисту інформації, формулювати їх математичну постановку та обирати раціональний метод вирішення.

РН 13. Впроваджувати, налаштовувати, супроводжувати та підтримувати функціонування програмних і програмно-апаратних комплексів і систем кібербезпеки та захисту інформації як необхідні процедури для функціонування інформаційних й інформаційно-комунікаційних систем та/або інфраструктури організації в цілому.

Заплановані результати навчання за навчальною дисципліною

Знання:

- різновидів алгоритмів і основних засобів їхньої побудови;
- елементів теорії алгоритмів;
- загальних принципів побудови алгоритмів;
- етапів розв'язування задач за допомогою комп'ютера;
- концепції структурного програмування;
- основних алгоритмічних конструкцій (послідовність, розгалуження, цикл) та відповідних їм команд мовою програмування C++;
- концепцію типів даних і операції над даними різних типів;
- елементів алгоритмічної мови програмування C++ (алфавіт мови, типи даних, правила записування арифметичних виразів);
- особливостей створення програмних проектів засобами C++;
- особливостей програмного опрацювання масивів, символьних даних (рядків), файлів;
- різновидів динамічних структур даних;
- алгоритми пошуку даних в послідовностях, строках та інших структурах;
- алгоритми сортування даних.

Уміння:

- працювати з програмним забезпеченням і файловою системою, проводити найпростіші операції щодо обслуговування комп'ютера, адекватно і обґрунтовано вибирати програмний засіб для розв'язання фахових завдань і здійснювати обмін даними між програмами;
- створювати лінійні, розгалужені і циклічні алгоритми з використанням простих і структурованих типів даних для розв'язування задач;
- будувати і описувати блок-схеми;
- формалізувати прикладну задачу та інтерпретувати її в термінах програмування;
- розробляти алгоритми розв'язування типових математичних та прикладних задач;
- обирати адекватний завданню метод реалізації типових алгоритмів оброблення даних;
- розробляти програми з лінійною, розгалуженою та циклічною структурами мовою C++;
- розробляти програмні проекти для опрацювання масивів, рядків та файлів мовою C++;
- самостійно опановувати нові методи та технології розробки програм;
- розв'язувати задачі, пов'язані з пошуком та сортуванням даних;
- розробляти програмні проекти для створення та опрацювання динамічних структур даних.

Навички:

- застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук;
- використовувати сучасний математичний апарат неперервного та дискретного аналізу, лінійної алгебри, аналітичної геометрії, в професійній діяльності для розв'язання задач теоретичного та прикладного характеру в процесі проектування та реалізації об'єктів інформатизації;
- проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій;
- використовувати методи чисельного диференціювання та інтегрування функцій, розв'язання звичайних диференціальних та інтегральних рівнянь, особливостей чисельних методів та можливостей їх адаптації до інженерних задач, мати навички програмної реалізації чисельних методів;
- розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук;
- володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проекту документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. Алфавіт мови C++. Типи даних. Змінні і константи. Правила записування арифметичних виразів: операнди і вирази, арифметичні операції, оператори присвоювання, зведення типів, математичні функції, функції перетворення числових типів. Введення-виведення. Види базових алгоритмів. Програмування лінійних алгоритмів.

Тема 2. Програмування розгалужених алгоритмів. Операції відношення та логічні операції. Умовний оператор if. Тернарна умовна операція. Оператор вибору варіантів switch. Оператор безумовного переходу go to.

Тема 3. Програмування циклів. Оператор циклу з параметром for. Алгоритми та програми обчислення сум, добутків. Вкладені цикли. Табулювання функцій. Опрацювання послідовностей введених чисел.

Тема 4. Оператори циклу з попередньою і наступною умовами while, do while. Обчислення сум скінченного та нескінченного числа доданків.

Тема 5. Правила організації функцій. Локальні і глобальні змінні. Формальні та фактичні параметри. Способи передавання результатів до основної програми. Оператор return.

Тема 6. Поняття масиву. Одновимірні масиви: оголошення, введення/виведення елементів, програмування базових алгоритмів опрацювання одновимірних масивів.

Тема 7. Опрацювання одновимірних масивів у функціях.

Тема 8. Двовимірні масиви (матриці): оголошення, введення/виведення елементів, програмування базових алгоритмів опрацювання двовимірних масивів.

Тема 9. Опрацювання матриць у функціях.

Тема 10. Вказівники. Вказівники на одновимірні масиви. Арифметика вказівників.

Тема 11. Поняття статичної і динамічної пам'яті. Динамічні масиви

Тема 12. Символьний тип даних. Функції C++ для роботи з символами. Програмне опрацювання символьних масивів.

Тема 13. Рядки char*. Функції C++ для роботи з рядками. Програмне опрацювання рядків.

Тема 14. Оголошення та звернення до елементів структур. Приклад програми зі структурою.

Тема 15. Текстові файли: функції для опрацювання текстових файлів, записування і зчитування текстових файлів.

Тема 16. Бінарні файли: функції для опрацювання бінарних файлів, записування і зчитування бінарних файлів.

Тема 17. Алгоритми послідовного і бінарного пошуку.

Тема 18. Алгоритм текстового пошуку Кнута-Морріса-Пратта.

Тема 19. Алгоритми сортування масивів: бульбашковий, вибором, вставкою, Шела.

Тема 20. Лінійні списки. Створення, перегляд, стандартні алгоритми опрацювання елементів списку.

Тема 21. Вставлення і видалення елементів лінійних списків.

Тема 22. Різновиди списків: двозв'язні списки, циклічні списки.

Тема 23. Черга і стек.

Тема 24. Рекурсія. Бінарні дерева.

Тема 25. Бінарні дерева. Алгоритми пошуку на деревах.

Тема 26. Графи. Матриці та списки суміжності.

Тема 27. Графи. Алгоритм пошуку найкоротшого шляху.

Тема 28. Графи. Алгоритми обходу графа у ширину та глибину.

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин									
	Денна форма					Заочна форма				
	Усього	у тому числі				Усього	у тому числі			
		Лекц.	Прак.	Лаб.	Сам. роб.		Лекц.	Прак.	Лаб.	Сам. роб.
Тема 1. Алфавіт мови C++. Типи даних. Змінні і константи. Правила записування арифметичних виразів: операнди і вирази, арифметичні операції, оператори присвоювання, зведення типів, математичні функції, функції перетворення числових типів. Введення-виведення. Види базових алгоритмів. Програмування лінійних алгоритмів.	10	2	2		6	10	2			8
Тема 2. Програмування розгалужених алгоритмів. Операції відношення та логічні операції. Умовний оператор if. Тернарна умовна операція. Оператор вибору варіантів switch. Оператор безумовного переходу go to.	10	2	2		6	10		2		8
Тема 3. Програмування циклів. Оператор циклу з параметром for. Алгоритми та програми обчислення сум, добутків. Вкладені цикли. Табулювання функцій. Опрацювання послідовностей введених чисел.	10	2	2		6	10	2			8
Тема 4. Оператори циклу з попередньою і наступною умовами while, do while. Обчислення сум скінченного та нескінченного числа доданків.	10	2	2		6	10				10
Тема 5. Правила організації	10	2	2		6	10				10

функцій. Локальні і глобальні змінні. Формальні та фактичні параметри. Способи передавання результатів до основної програми. Оператор return.										
Тема 6. Поняття масиву. Одновимірні масиви: оголошення, введення/виведення елементів, програмування базових алгоритмів опрацювання одновимірних масивів.	10	2	2		6	10		2		8
Тема 7. Опрацювання одновимірних масивів у функціях.	10	2	2		6	10				10
Тема 8. Двовимірні масиви (матриці): оголошення, введення/виведення елементів, програмування базових алгоритмів опрацювання двовимірних масивів.	10	2	2		6	10		2		8
Тема 9. Опрацювання матриць у функціях.	10	2	2		6	10	2			8
Тема 10. Вказівники. Вказівники на одновимірні масиви. Арифметика вказівників.	10	2	2		6	10				10
Тема 11. Поняття статичної і динамічної пам'яті. Динамічні масиви	10	2	2		6	10				10
Тема 12. Символьний тип даних. Функції C++ для роботи з символами. Програмне опрацювання символьних масивів.	10	2	2		6	10		2		8
Тема 13. Рядки char*. Функції C++ для роботи з рядками. Програмне опрацювання рядків.	10	2	2		6	10				10
Тема 14. Оголошення та	10	2	2		6	10				10

звернення до елементів структур. Приклад програми зі структурою.										
Тема 15. Текстові файли: функції для опрацювання текстових файлів, записування і зчитування текстових файлів.	10	2	2		6	10				10
Тема 16. Бінарні файли: функції для опрацювання бінарних файлів, записування і зчитування бінарних файлів.	10	2	2		6	10				10
Тема 17. Алгоритми послідовного і бінарного пошуку.	10	2	2		6	12	2			10
Тема 18. Алгоритм текстового пошуку Кнута-Морріса-Пратта.	10	2	2		6	8				8
Всього	180	36	36	0	108	180	8	8	0	164
Підсумковий контроль - залік										

Назви змістових модулів і тем	Кількість годин									
	Денна форма					Заочна форма				
	Усього	у тому числі				Усього	у тому числі			
		Лекц.	Прак.	Лаб.	Сам. роб.		Лекц.	Прак.	Лаб.	Сам. роб.
Тема 19. Алгоритми сортування масивів: бульбашковий, вибором, вставкою, Шела.	12	2	4		6	12				12
Тема 20. Лінійні списки. Створення, перегляд, стандартні алгоритми опрацювання елементів списку.	12	2	2		8	14		2		12
Тема 21. Вставлення і видалення елементів лінійних списків.	12	2	4		6	12				12
Тема 22. Різновиди списків: двозв'язні списки, циклічні	12	2	2		8	12	2			10

списки.										
Тема 23. Черга і стек.	12	2	2		8	12				12
Тема 24. Рекурсія. Бінарні дерева.	12	2	2		8	10				10
Тема 25. Бінарні дерева. Алгоритми пошуку на деревах.	12	2	4		6	12				12
Тема 26. Графи. Матриці та списки суміжності.	12	4	2		6	12		2		10
Тема 27. Графи. Алгоритм пошуку найкоротшого шляху.	12	4	2		6	12	2			10
Тема 28. Графи. Алгоритми обходу графа у ширину та глибину.	12	4	2		6	12				12
Всього	120	26	26	0	68	120	4	4	0	112
РАЗОМ	300	62	62	0	176	300	12	12	0	276
Підсумковий контроль - екзамен										

5. ПИТАННЯ ДО ПРАКТИЧНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин	
		денна форма	заочна форма
1	Тема 1. 1. Напишіть програму, яка зчитує з клавіатури три дійсних числа a, b, c та обчислює значення виразу: $(a^2 + b) / (c - 1.5) + \sqrt{ a }$. Виведіть результат з точністю до 4 знаків після коми. Використовуйте константу $\pi = 3.14159$. 2. Створіть програму, яка запитує у користувача радіус кола (тип double), обчислює його площу та довжину кола. Використовуйте константу для π та функцію pow або * для квадрата. 3. Напишіть лінійну програму: зчитайте ціле число n (секунди), переведіть його у години, хвилини та секунди та виведіть у форматі «Г:ХВ:С» (наприклад, 3665 → 1:1:5). Використовуйте операції / та %. 4. Створіть програму, яка зчитує два цілих числа x та y, обчислює їх суму, різницю, добуток, частку (з плаваючою точкою) та залишок від ділення. Виведіть усі результати в акуратному вигляді.	2	
2	Тема 2. 1. Напишіть програму, яка зчитує число x та виводить повідомлення: «Додатне», «Від'ємне» або «Нуль». Використовуйте if-else. 2. Зчитайте три сторони трикутника (a, b, c). Визначте, чи існує такий трикутник (нерівність трикутника), і якщо так — чи є він рівнобедреним, рівностороннім чи різностороннім. Використовуйте вкладені if. 3. Створіть програму-калькулятор: зчитайте два числа та символ операції (+, -, *, /). Виконайте відповідну операцію за допомогою switch. Обробіть ділення на нуль та невідомий символ.	2	2

	4. Напишіть програму, яка зчитує оцінку (1–12) та виводить словесну характеристику: 10–12 — «Відмінно», 7–9 — «Добре», 4–6 — «Задовільно», 1–3 — «Незадовільно». Використовуйте тернарний оператор або switch.		
3	Тема 3. 1. Напишіть програму, яка обчислює суму всіх парних чисел від 1 до n (n вводиться з клавіатури) за допомогою циклу for. 2. Створіть програму для табулювання функції $y = x^2 - 3x + 5$ на інтервалі [a; b] з кроком h. Виведіть таблицю x y у два стовпці. 3. Напишіть програму, яка виводить прямокутний трикутник з символів '*' висотою n (n вводиться). Використовуйте вкладені цикли for. 4. Зчитуйте послідовність чисел з клавіатури (закінчення — введення 0). Обчисліть кількість додатних, від'ємних та нульових чисел.	2	
4	Тема 4. 1. Напишіть програму, яка зчитує числа з клавіатури, доки не буде введено від'ємне число. Обчисліть середнє арифметичне введених додатних чисел (використовуйте while). 2. Обчисліть суму ряду $1 + 1/2 + 1/3 + \dots + 1/n$, де n вводиться користувачем (використовуйте for або while). 3. Напишіть програму для обчислення суми ряду з точністю ϵ (ϵ вводиться): $S = 1 - 1/3! + 1/5! - 1/7! + \dots$ (знаменник — непарні факторіали). Використовуйте do-while. 4. Створіть програму, яка обчислює наближене значення $\sqrt{2}$ методом Ньютона ($x_{n+1} = (x_n + 2/x_n)/2$) з точністю ϵ . Початкове наближення $x_0 = 1$.	2	
5	Тема 5. 1. Напишіть функцію <code>double power(double a, int n)</code> , яка обчислює a^n ($n \geq 0$). Використовуйте цикл. Викличте її в main для введених значень. 2. Створіть функцію <code>bool isPrime(int x)</code> , яка повертає true, якщо число просте. У main виведіть усі прості числа від 2 до 100. 3. Напишіть функцію <code>void swap(int &x, int &y)</code> , яка міняє місцями два числа за посиланням. Продемонструйте роботу в main. 4. Створіть функцію <code>int maxOfThree(int a, int b, int c)</code> , яка повертає найбільше з трьох чисел. Використовуйте тернарний оператор всередині функції.	2	
6	Тема 6. 1. Напишіть програму: зчитайте розмір масиву n, потім n чисел. Виведіть масив у прямому та зворотному порядку. 2. У масиві з n елементів знайдіть максимальний елемент та його індекс. Виведіть значення та позицію. 3. Створіть програму, яка заповнює масив випадковими числами від -20 до 30 (використовуйте <code>rand()</code>), потім обчислює суму від'ємних елементів. 4. Напишіть програму: зчитайте масив, підрахуйте кількість пар сусідніх елементів, сума яких парна.	2	2
7	Тема 7. 1. Напишіть функцію <code>void printArray(int arr[], int size)</code> , яка виводить масив у рядок з пробілами. 2. Створіть функцію <code>int sumPositive(int arr[], int n)</code> , яка повертає суму додатних елементів масиву. 3. Напишіть функцію <code>void sortAscending(int arr[], int n)</code> , яка сортує масив за зростанням методом бульбашки (без використання <code>std::sort</code>). 4. Реалізуйте функцію <code>int findMinIndex(const int arr[], int size)</code> , яка повертає індекс мінімального елемента. Продемонструйте виклик у main.	2	
	Всього	14	4

№ з/п	Назва теми	Кількість годин	
		денна форма	заочна форма
1	Тема 8. 1. Напишіть програму, яка зчитує розмір квадратної матриці n ($n \leq 10$), потім елементи матриці рядок за рядком. Виведіть матрицю у вигляді акуратної таблиці з вирівнюванням. 2. У матриці розміром $m \times n$ знайдіть суму елементів на головній діагоналі та на побічній діагоналі. Виведіть обидві суми. 3. Створіть програму: заповніть матрицю 5×5 випадковими числами від -50 до 50, потім виведіть її та обчисліть суму кожного рядка (виведіть масив сум рядків). 4. Напишіть програму для транспонування матриці (заміна рядків на стовпці). Виведіть початкову та транспоновану матрицю.	2	2
2	Тема 9. 1. Напишіть функцію <code>void printMatrix(int mat[][10], int rows, int cols)</code> , яка виводить матрицю з фіксованим розміром стовпців. 2. Створіть функцію <code>int sumDiagonal(const int mat[][MAX], int n)</code> , яка повертає суму елементів головної діагоналі квадратної матриці. 3. Напишіть функцію <code>void transpose(int mat[][10], int m, int n)</code> , яка транспонує матрицю (потрібно передати копію або використовувати додаткову матрицю). 4. Реалізуйте функцію <code>double averageRow(const int mat[][MAX], int row, int cols)</code> , яка повертає середнє арифметичне елементів заданого рядка. Продемонструйте виклик для всіх рядків.	2	
3	Тема 10. 1. Напишіть програму: оголошіть масив з 10 елементів, ініціалізуйте вказівник на його початок. Виведіть елементи масиву за допомогою арифметики вказівників (<code>p++</code> , <code>*(p+i)</code>). 2. Створіть функцію <code>void swapByPointer(int *a, int *b)</code> , яка міняє місцями два числа через вказівники. Продемонструйте в <code>main</code> . 3. Напишіть програму, яка за допомогою вказівника знаходить максимальний елемент у масиві та його адресу. Виведіть значення та адресу. 4. Використовуючи арифметику вказівників, скопіюйте один масив у інший без використання індексів <code>[]</code> .	2	
4	Тема 11. 1. Напишіть програму: запитуйте у користувача розмір масиву n , створіть динамічний масив за допомогою <code>new[]</code> , заповніть випадковими числами, виведіть, потім видаліть за допомогою <code>delete[]</code> . 2. Створіть динамічний масив для зберігання n чисел, обчисліть їх середнє арифметичне та дисперсію. 3. Напишіть програму з динамічним двовимірним масивом (матрицею $m \times n$): виділіть пам'ять через <code>new[]</code> , заповніть, виведіть, звільніть пам'ять правильно (цикл + <code>delete[]</code>). 4. Реалізуйте функцію <code>int* createArray(int size)</code> , яка повертає вказівник на динамічний масив, заповнений числами від 1 до <code>size</code> .	2	
5	Тема 12. 1. Напишіть програму, яка зчитує символ з клавіатури та визначає: це велика літера, мала літера, цифра чи інший символ (використовуйте <code>isupper</code> , <code>islower</code> , <code>isdigit</code> з <code><cctype></code>). 2. Створіть символьний масив <code>char str[100]</code> . Зчитайте рядок за допомогою <code>cin.getline</code> , підрахуйте кількість голосних літер (а, е, и, і, о, у, я, ю, є, ї). 3. Напишіть програму, яка перетворює введений рядок у верхній регістр (використовуйте <code>toupper</code>) та виводить його.	2	2

	4. Зчитайте символи в масив, поки не буде введено крапку. Виведіть кількість введених символів та сам рядок.		
6	Тема 13. 1. Напишіть програму: зчитайте два рядки за допомогою <code>cin.getline</code> , порівняйте їх за допомогою <code>strcmp</code> та виведіть, який більший (або рівні). 2. Створіть програму, яка зчитує рядок та підраховує кількість слів (слова розділені пробілами, використовуйте <code>strlen</code> та цикл). 3. Реалізуйте функцію <code>void reverseString(char str[])</code> , яка перевертає рядок на місці (без використання додаткового масиву). 4. Напишіть програму, яка замінює всі пробіли в рядку на символ <code>'_'</code> (використовуйте <code>strcpy</code> , <code>strcat</code> або цикл).	2	
7	Тема 14. 1. Оголошіть структуру <code>Student { char name[50]; int age; double average; }</code> . Заповніть одну структуру з клавіатури та виведіть її поля. 2. Створіть масив структур <code>Student</code> розміром 3. Зчитайте дані про трьох студентів та виведіть студента з найвищою середньою оцінкою. 3. Напишіть структуру <code>Point { double x, y; }</code> . Створіть функцію <code>double distance(Point a, Point b)</code> , яка обчислює відстань між двома точками. 4. Реалізуйте структуру <code>Date { int day, month, year; }</code> . Напишіть функцію <code>bool isLeapYear(int year)</code> та перевірте, чи високосний рік введеної дати.	2	
8	Тема 15. 1. Напишіть програму, яка записує в текстовий файл <code>"output.txt"</code> числа від 1 до 20, кожне з нового рядка. 2. Створіть програму, яка зчитує текстовий файл <code>"input.txt"</code> та виводить його вміст на екран. 3. Напишіть програму: зчитайте з клавіатури рядки (до порожнього рядка), запишіть їх у файл <code>"notes.txt"</code> . 4. Обчисліть кількість рядків та символів у текстовому файлі (використовуйте <code>ifstream</code> , <code>getline</code>).	2	
9	Тема 16. 1. Оголошіть структуру <code>Book { char title[100]; int pages; float price; }</code> . Запишіть 3 книги в бінарний файл <code>"books.dat"</code> за допомогою <code>fwrite</code> . 2. Прочитайте бінарний файл <code>"books.dat"</code> та виведіть інформацію про всі книги (<code>fread</code>). 3. Додайте в кінець бінарного файлу ще одну структуру (використовуйте <code>ios::app</code>). 4. Напишіть програму для пошуку книги за назвою в бінарному файлі (послідовне читання).	2	
10	Тема 17. 1. Напишіть функцію <code>int linearSearch(int arr[], int n, int key)</code> , яка повертає індекс першого входження ключа або -1. 2. Реалізуйте бінарний пошук <code>int binarySearch(int arr[], int n, int key)</code> (масив має бути відсортованим). 3. Створіть програму: заповніть масив випадковими числами, відсортуйте (можна <code>std::sort</code>), знайдіть елемент послідовним та бінарним пошуком, порівняйте кількість порівнянь. 4. Напишіть програму для пошуку мінімального елемента в масиві за допомогою лінійного пошуку.	2	
11	Тема 18. 1. Напишіть функцію для обчислення префікс-функції (π -масиву) для рядка-шаблону. 2. Реалізуйте алгоритм КМП: знайдіть усі входження підрядка <code>pattern</code> у рядку <code>text</code> . 3. Створіть програму: зчитайте текст та шаблон, виведіть позиції всіх	2	

	знайдених входжень за допомогою КМП. 4. Порівняйте ефективність КМП та наївного пошуку на довгому тексті (наприклад, 1000 символів).		
12	Тема 19. 1. Реалізуйте бульбашкове сортування (bubble sort) та виведіть масив після кожного проходу (для демонстрації). 2. Напишіть сортування вибором (selection sort): знайдіть мінімум у невідсортованій частині та обміняйте. 3. Реалізуйте сортування вставками (insertion sort) для масиву з 20 елементів. 4. Напишіть сортування Шелла (shell sort) з кроками 5, 3, 1 та протестуйте на випадковому масиві.	2	
13	Тема 20. 1. Оголошіть структуру Node { int data; Node* next; }. Створіть список з 5 елементів (вставка в кінець). 2. Напишіть функцію void printList(Node* head), яка виводить усі елементи списку.	2	2
14	Тема 21. 1. Реалізуйте вставку елемента на початок списку та після заданого значення. 2. Напишіть функцію для видалення першого входження заданого значення зі списку.	4	
15	Тема 22. 1. Оголошіть структуру для двозв'язного списку (prev та next). Додайте 4 елементи. 2. Напишіть функцію для виведення двозв'язного списку в прямому та зворотному напрямку. 3. Реалізуйте циклічний список: вставте елементи так, щоб останній вказував на перший. 4. Видаліть елемент з двозв'язного списку, правильно оновивши prev/next сусідів.	2	
16	Тема 23. 1. Реалізуйте стек на базі масиву: функції push, pop, top, isEmpty. 2. Реалізуйте чергу на базі масиву (з циклічним буфером): enqueue, dequeue, front. 3. Напишіть програму, яка використовує стек для перевірки балансу дужок у виразі. 4. Використовуйте чергу для моделювання черги друку (додайте 5 завдань та обробіть їх).	2	
17	Тема 24. 1. Напишіть рекурсивну функцію для обчислення факторіалу та чисел Фібоначчі. 2. Оголошіть структуру TreeNode { int data; TreeNode* left, *right; }. Створіть просте бінарне дерево.	2	
18	Тема 25. 1. Напишіть рекурсивні функції для обходу дерева: pre-order, in-order, post-order. 2. Реалізуйте рекурсивний пошук елемента в бінарному дереві пошуку (BST).	4	
19	Тема 26. 1. Представте неорієнтований граф за допомогою матриці суміжності 5×5. Виведіть її. 2. Реалізуйте список суміжності для графа (використовуйте vector<vector<int>>).	2	2

20	Тема 27. 1. Напишіть функцію для обходу графа в глибину (DFS) рекурсивно. 2. Реалізуйте обхід у ширину (BFS) за допомогою черги.	2	
21	Тема 28. 1. Напишіть програму для пошуку найкоротшого шляху за алгоритмом Дейкстри (використовуйте пріоритетну чергу або простий варіант). 2. Знайти шлях від вершини s до t за допомогою BFS (відновлення шляху через parent[]).	2	
	Всього	48	8

6. САМОСТІЙНА РОБОТА

До самостійної роботи здобувачів щодо вивчення дисципліни «**Основи програмування**» включаються наступні тематики завдання.

1. Знайомство з науковою та навчальною літературою відповідно зазначених у програмі тем.
2. Опрацювання лекційного матеріалу.
3. Підготовка до практичних занять.
4. Консультації з викладачем протягом семестру.
5. Самостійне опрацювання окремих питань навчальної дисципліни.
6. Підготовка до підсумкового контролю.

Тематика та питання до самостійної підготовки та індивідуальних завдань

№ з/п	Назва теми	Кількість годин	
		денна форма	заочна форма
1	Тема 1. Алфавіт мови C++. Типи даних. Змінні і константи. Правила записування арифметичних виразів: операнди і вирази, арифметичні операції, оператори присвоювання, зведення типів, математичні функції, функції перетворення числових типів. Введення-виведення. Види базових алгоритмів. Програмування лінійних алгоритмів.	6	8
2	Тема 2. Програмування розгалужених алгоритмів. Операції відношення та логічні операції. Умовний оператор if. Тернарна умовна операція. Оператор вибору варіантів switch. Оператор безумовного переходу go to.	6	8
3	Тема 3. Програмування циклів. Оператор циклу з параметром for. Алгоритми та програми обчислення сум, добутків. Вкладені цикли. Табулювання функцій. Опрацювання послідовностей введених чисел.	6	8
4	Тема 4. Оператори циклу з попередньою і наступною умовами while, do while. Обчислення сум скінченного та нескінченного числа доданків.	6	10

5	Тема 5. Правила організації функцій. Локальні і глобальні змінні. Формальні та фактичні параметри. Способи передавання результатів до основної програми. Оператор return.	6	10
6	Тема 6. Поняття масиву. Одновимірні масиви: оголошення, введення/виведення елементів, програмування базових алгоритмів опрацювання одновимірних масивів.	6	8
7	Тема 7. Опрацювання одновимірних масивів у функціях.	6	10
8	Тема 8. Двовимірні масиви (матриці): оголошення, введення/виведення елементів, програмування базових алгоритмів опрацювання двовимірних масивів.	6	8
9	Тема 9. Опрацювання матриць у функціях.	6	8
10	Тема 10. Вказівники. Вказівники на одновимірні масиви. Арифметика вказівників.	6	10
11	Тема 11. Поняття статичної і динамічної пам'яті. Динамічні масиви	6	10
12	Тема 12. Символьний тип даних. Функції C++ для роботи з символами. Програмне опрацювання символьних масивів.	6	8
13	Тема 13. Рядки char*. Функції C++ для роботи з рядками. Програмне опрацювання рядків.	6	10
14	Тема 14. Оголошення та звернення до елементів структур. Приклад програми зі структурою.	6	10
15	Тема 15. Текстові файли: функції для опрацювання текстових файлів, записування і зчитування текстових файлів. Тема	6	10
16	Тема 16. Бінарні файли: функції для опрацювання бінарних файлів, записування і зчитування бінарних файлів.	6	10
17	Тема 17. Алгоритми послідовного і бінарного пошуку.	6	10
18	Тема 18. Алгоритм текстового пошуку Кнута-Морріса-Пратта.	6	8
	Всього	108	164

№ з/п	Назва теми	Кількість годин	
		денна форма	заочна форма
1	Тема 19. Алгоритми сортування масивів: бульбашковий, вибором, вставкою, Шела.	6	12
2	Тема 20. Лінійні списки. Створення, перегляд, стандартні алгоритми	8	12

	опрацювання елементів списку.		
3	Тема 21. Вставлення і видалення елементів лінійних списків.	6	12
4	Тема 22. Різновиди списків: двозв'язні списки, циклічні списки.	8	10
5	Тема 23. Черга і стек. Тема	8	12
6	Тема 24. Рекурсія. Бінарні дерева.	8	10
7	Тема 25. Бінарні дерева. Алгоритми пошуку на деревах.	6	12
8	Тема 26. Графи. Матриці та списки суміжності.	6	10
9	Тема 27. Графи. Алгоритм пошуку найкоротшого шляху.	6	10
10	Тема 28. Графи. Алгоритми обходу графа у ширину та глибину.	6	12
	Всього	68	112

7. ВИДИ ТА МЕТОДИ КОНТРОЛЮ

Види контролю	Складові оцінювання	
	Для екзамену	Для заліку
поточний контроль , який здійснюється у ході: проведення практичних (лабораторних) занять, виконання самостійної роботи, індивідуальних завдань, дослідна робота здобувача тощо.	50%	100%
підсумковий контроль	50%	

Методи діагностики знань (контролю)	Фронтальне опитування, практичні завдання, індивідуальні завдання, робота у групах, розв'язання практичних завдань, залік, екзамен.
-------------------------------------	---

Питання до екзамену

1. Назвіть алфавіт мови C++, основні типи даних (цілі, дійсні, символьні, логічні). Як оголошуються змінні та константи? Наведіть приклади.
2. Які арифметичні операції та оператори присвоювання існують у C++? Поясніть пріоритет операцій, зведення типів та явне приведення типів.
3. Опишіть функції введення-виведення. Наведіть приклад програми для читання/виведення чисел з клавіатури.
4. Що таке базові алгоритми? Наведіть приклади лінійних алгоритмів (обчислення виразу, знаходження максимуму з трьох чисел).
5. Поясніть роботу умовного оператора if...else, вкладених if. Наведіть приклад.
6. Як працює оператор вибору switch? Які особливості використання? Порівняйте з if.
7. Наведіть синтаксис оператора циклу for. Напишіть програму для обчислення суми перших n натуральних чисел.
8. Опишіть вкладені цикли for. Наведіть приклад табулювання функції (наприклад, $y = \sin(x)$ на інтервалі).
9. Як опрацьовувати послідовність введених чисел до введення нуля або до EOF?
10. Порівняйте цикли while та do...while. Наведіть приклад обчислення суми нескінченної

послідовності з точністю ε .

11. Як оголошуються функції в C++? Поясніть локальні та глобальні змінні, область видимості.
12. Що таке формальні та фактичні параметри? Опишіть передачу за значенням та за посиланням. Наведіть приклад.
13. Як працює оператор return? Які способи повернення результату з функції (return, через параметр-посилання)?
14. Що таке масив? Як оголошуються одновимірні масиви? Наведіть приклади ініціалізації та введення/виведення елементів.
15. Напишіть базові алгоритми опрацювання одновимірних масивів: знаходження max/min, суми, середнього, кількості додатних елементів.
16. Як передавати одновимірні масиви в функції? Поясніть різницю між передачею масиву та вказівника на масив.
17. Опишіть оголошення та ініціалізацію двовимірних масивів (матриць). Як вводити/виводити матрицю?
18. Напишіть алгоритми опрацювання матриць: сума елементів, сума головної/побічної діагоналі, транспонування.
19. Як передавати матриці в функції? Поясніть використання подвійних вказівників або фіксованих розмірів.
20. Що таке вказівник? Як оголошується вказівник? Наведіть арифметику вказівників.
21. Поясніть вказівники на масиви. Як використовувати вказівник для проходження по масиву замість індексів?
22. Що таке статична та динамічна пам'ять? Як створювати/видаляти динамічні масиви? Наведіть приклад.
23. Опишіть символічний тип char. Які функції бібліотеки <cctype> для роботи з символами?
24. Що таке рядок char* або char[]? Наведіть функції роботи з рядками.
25. Як опрацьовувати рядки: підрахунок голосних, заміна символів, перевірка паліндрому?
26. Що таке структура (struct)? Як оголошувати, ініціалізувати, звертатися до полів структури? Наведіть приклад.
27. Опишіть роботу з текстовими файлами: fopen/fclose, fprintf/fscanf, fgets, fputs. Наведіть приклад копіювання файлу.
28. Як працювати з бінарними файлами (fread, fwrite, fseek)? Наведіть приклад запису/читання структури в бінарний файл.
29. Порівняйте алгоритми послідовного та бінарного пошуку. У яких умовах бінарний пошук ефективніший?
30. Опишіть алгоритм Кнута-Морріса-Пратта (КМП) для текстового пошуку. Наведіть принцип побудови префікс-функції.
31. Порівняйте алгоритми сортування: бульбашковий, вибором, вставками, Шелла. Які їх складності (O-нотація)?
32. Що таке лінійний список? Як створювати, переглядати, вставляти/видаляти елементи в однозв'язному списку?
33. Поясніть двозв'язні та циклічні списки. Наведіть переваги та приклади операцій.
34. Що таке стек та черга? Як реалізувати їх на базі масиву або списку? Наведіть приклади.
35. Що таке рекурсія? Наведіть приклад рекурсивного обчислення факторіалу або чисел Фібоначчі.
36. Опишіть бінарне дерево: структура вузла, обхід у глибину, пошук елемента.
37. Як представляти граф: матриця суміжності, список суміжності? Наведіть приклади.
38. Опишіть алгоритми обходу графа: у ширину та у глибину. Наведіть застосування.
39. Поясніть алгоритм пошуку найкоротшого шляху. Наведіть приклад.

8. ОЦІНЮВАННЯ ЗДОБУВАЧІВ ВИЩОЇ ТА ФАХОВОЇ ПЕРДВИЩОЇ ОСВІТИ ЗА НАВЧАЛЬНУ ДИСЦИПЛІНУ У МІЖНАРОДНОМУ ГУМАНІТАРНОМУ УНІВЕРСИТЕТІ

ЗАЛІК

<i>Поточний контроль</i>			
Види роботи	Планові терміни виконання	Форми контролю та звітності	Максимальний відсоток оцінювання
Систематичність і активність роботи на практичних (лабораторних) заняттях			
1.1. Підготовка до практичних (лабораторних) занять	Відповідно до робочої програми та розкладу занять	Перевірка обсягу та якості засвоєного матеріалу під час практичних (лабораторних) занять	60
Виконання завдань для самостійного опрацювання			
1.2. Індивідуальне тестування, завдання, підготовка реферату (есе) за заданою тематикою	Відповідно до робочої програми та розкладу занять	Розгляд відповідного матеріалу під час аудиторних занять або ІКР ¹ , перевірка конспектів навчальних текстів тощо	20
Виконання індивідуальних завдань, дослідна робота здобувача			
1.3. Інші види індивідуальних завдань, в т.ч. підготовка наукових публікацій, участь у роботі круглих столів, конференцій тощо.	Відповідно до робочої програми та розкладу занять	Обговорення результатів проведеної роботи під час аудиторних занять, наукових конференцій та круглих столів.	20
Разом балів за поточний контроль			100
<i>Загальний результат оцінювання</i>			100

Поточний контроль знань здобувачів освіти при формі контролю **залік** (максимум 100 балів) оцінюється за шкалою («2», «3», «4», «5»), з обов'язковим переведенням балів за кожний вид роботи таким чином:

- за підготовку до аудиторних занять (максимум 60 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості практичних (лабораторних) занять, для переведу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 12;
- за виконання завдань для самостійного опрацювання (тестування, завдання, підготовка реферату (есе) за заданою тематикою) (максимум 20 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості програмного матеріалу, що виноситься на самостійне вивчення, для переведу зазначених балів середньоарифметична оцінка множиться на коефіцієнт 4;
- за інші види індивідуальних завдань, в т.ч. підготовку наукових публікацій, участь у роботі круглих столів, конференцій тощо (максимум 20- балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості інших видів індивідуальних завдань, для переведу

¹ Індивідуально-консультативна робота викладача зі здобувачами

балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 4.

КРИТЕРІЇ ОЦІНЮВАННЯ ЗА ПОТОЧНИЙ КОНТРОЛЬ

- «2»- виставляється за неправильну відповідь (письмову роботу), яка не відповідає змісту теми та свідчить про нерозуміння її основних положень;
- «3» -виставляється за відповідь (письмову роботу), яка відповідає змісту теми, але є неповною і неточною у визначенні понять, свідчить про неповне розуміння основних положень навчального матеріалу, свідчить про те, що знання здобувача мають фрагментарний, поверховий характер;
- «4» - оцінюється правильна і обґрунтована відповідь (письмова робота), з якої видно, що здобувач знає й розуміє теоретичний матеріал в обсязі навчальної теми, володіє необхідними навичками, не допускає суттєвих помилок.
- «5» - оцінюється повна, правильна, ґрунтовна відповідь (письмова робота) на запитання теми, що передбачає логічність і послідовність викладу матеріалу, володіння термінологією, показує вміння повно й глибоко використовувати теоретичні знання в практичній площині.

ЕКЗАМЕН

<i>Поточний контроль</i>			
Види роботи	Планові терміни виконання	Форми контролю та звітності	Максимальний відсоток оцінювання
Систематичність і активність роботи на практичних (лабораторних) заняттях			
1.1. Підготовка до практичних (лабораторних) занять	Відповідно до робочої програми та розкладу занять	Перевірка обсягу та якості засвоєного матеріалу під час практичних (лабораторних) занять	30
Виконання завдань для самостійного опрацювання			
1.2. Індивідуальне тестування, завдання, підготовка реферату (есе) за заданою тематикою	Відповідно до робочої програми тарозкладу занять	Розгляд відповідного матеріалу під час аудиторних занять або ІКР ² , перевірка конспектів навчальних текстів тощо	10
Виконання індивідуальних завдань, дослідна робота здобувача			
1.3. Інші види індивідуальних завдань, в т.ч. підготовка наукових публікацій, участь у роботі круглих столів, конференцій тощо.	Відповідно до робочої програми тарозкладу занять	Обговорення результатів проведеної роботи під час аудиторних занять, наукових конференцій та круглих столів.	10
Разом балів за поточний контроль			50
Підсумковий контроль екзамен			50
Загальний результат оцінювання			100

Поточний контроль знань здобувачів освіти при формі контролю **екзамен** (максимум 50 балів) оцінюється за шкалою («2», «3», «4», «5»), з обов'язковим переведенням балів за кожний вид роботи таким чином:

² Індивідуально-консультативна робота викладача зі здобувачами

- за підготовку до аудиторних занять (максимум 30 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості практичних (лабораторних) занять, для переводу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 6;
- за виконання завдань для самостійного опрацювання (тестування, завдання, підготовка реферату (есе) за заданою тематикою) (максимум 10 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості програмного матеріалу, що вноситься на самостійне вивчення, для переводу зазначених балів середньоарифметична оцінка множиться на коефіцієнт 2;
- за інші види індивідуальних завдань, в т.ч. підготовку наукових публікацій, участь у роботі круглих столів, конференцій тощо (максимум 10- балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості інших видів індивідуальних завдань, для переводу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 2.

КРИТЕРІЇ ОЦІНЮВАННЯ ЗА ПОТОЧНИЙ КОНТРОЛЬ

- «2» - виставляється за неправильну відповідь (письмову роботу), яка не відповідає змісту теми та свідчить про нерозуміння її основних положень;
- «3» - виставляється за відповідь (письмову роботу), яка відповідає змісту теми, але є неповною і неточною у визначенні понять, свідчить про неповне розуміння основних положень навчального матеріалу, свідчить про те, що знання здобувача мають фрагментарний, поверховий характер;
- «4» - оцінюється правильна і обґрунтована відповідь (письмова робота), з якої видно, що здобувач знає й розуміє теоретичний матеріал в обсязі навчальної теми, володіє необхідними навичками, не допускає суттєвих помилок.
- «5» - оцінюється повна, правильна, ґрунтовна відповідь (письмова робота) на запитання теми, що передбачає логічність і послідовність викладу матеріалу, володіння термінологією, показує вміння повно й глибоко використовувати теоретичні знання в практичній площині.

КРИТЕРІЇ ОЦІНЮВАННЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ ЗДОБУВАЧІВ ЗА ІСПИТ

(максимум 50 балів)

Рівень знань оцінюється:

- від 40 до 50 балів - здобувач виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки;
- 30 до 40 балів - здобувач володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні відповіді;
- 20 - 30 балів - здобувач відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки.
- 10 - 20 балів - здобувач володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих;
- 0 - 10 балів - володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки.

9. КРИТЕРІЇ ЗАГАЛЬНИХ РЕЗУЛЬТАТІВ ОЦІНЮВАННЯ ЗНАНЬ ЗДОБУВАЧІВ (для екзамену / заліку)

Загальні результати знань здобувачів по даній дисципліні оцінюються наступним чином:

- **«Відмінно»/«зараховано» (А)** – від 90 до 100 балів. Здобувач виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки. Був присутній на лекціях та семінарських заняттях, під час яких давав вичерпні, обґрунтовані, теоретично і практично правильні відповіді, має конспект з виконаними завданнями до самостійної роботи, презентував реферат за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;
- **«Добре»/«зараховано» (В)** – від 82 до 89 балів. Здобувач володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні відповіді. Був присутній на лекціях та семінарських заняттях, має конспект з виконаними завданнями до самостійної роботи, презентував реферат за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;
- **«Добре»/«зараховано» (С)** – від 74 до 81 балів. Здобувач відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи, реферату та активність у науково-дослідній роботі;
- **«Задовільно»/«зараховано» (D)** – від 64 до 73 балів. Здобувач був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи та рефератів;
- **«Задовільно»/«зараховано» (E)** – від 60 до 63 балів. Здобувач був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки, має неповний конспект з завданнями до самостійної роботи.
- **«Незадовільно з можливістю повторного складання»/«не зараховано» (F_X)** – від 35 до 59 балів. Здобувач володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу.
- **«Незадовільно з обов'язковим повторним вивченням дисципліни»/«не зараховано» (F)** – від 1 до 34 балів. Здобувач не володіє навчальним матеріалом.

Таблиця відповідності результатів контролю знань за різними шкалами

100-бальною шкалою	Шкала за ECTS	За національною шкалою	
		екзамен	залік
90-100	A	Відмінно	Зараховано
82-89	B	Добре	Зараховано
74-81	C		
64-73	D	Задовільно	Зараховано
60-63	E		
35-59	F _x	Незадовільно	Не зараховано
1-34	F		

10. ДОТРИМАННЯ ВИМОГ АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

Викладання та засвоєння навчальної дисципліни передбачає неухильне дотримання вимог академічної доброчесності, яка регулюється Законом України “Про освіту” (Відомості Верховної Ради (ВВР), 2014, № 37-38, ст.2004), «Методичними рекомендаціями для закладів вищої освіти з підтримки принципів академічної доброчесності» (Лист МОН № 1/9-650 від 23.10.18 року), іншими нормативними документами. Зокрема, основні засади дотримання академічної доброчесності визначено у «Положенні про академічну доброчесність у Міжнародному гуманітарному університеті», «Кодексі академічної доброчесності Міжнародного гуманітарного університету» та «Інструкції щодо процедури технічної перевірки на наявність текстових запозичень (академічного плагіату)», затвердженої наказом МГУ від 03.05.2024 р., № 708а.

Відповідно до ст.42 Закону України “Про освіту” академічна доброчесність - це сукупність етичних принципів та визначених законом правил, якими мають керуватися учасники освітнього процесу під час навчання, викладання та провадження наукової (творчої) діяльності з метою забезпечення довіри до результатів навчання та/або наукових (творчих) досягнень.

Дотримання академічної доброчесності передбачає:

- 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);
- посилення на джерела інформації у разі використання ідей, розробок, тверджень, відомостей;
- дотримання норм законодавства про авторське право і суміжні права; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використанні методики досліджень і джерела інформації.

11. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. C++. Алгоритмізація та програмування : підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логінова, О.В. Задерейко. 2-ге вид. перероб. і доповн. Одеса : Фенікс, 2019. 477 с.
2. C++. Теорія та практика: навч. посібник з грифом МОНУ/ [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.] ; за ред. О. Г. Трофименко. Одеса : ВЦ ОНАЗ, 2011. 587 с.
3. C++. Основи програмування. Теорія та практика: підручник / [О.Г. Трофименко, Ю.В. Прокоп, І.Г.Швайко, Л.М. Буката та ін.] ; за ред. О.Г. Трофименко. Одеса: Фенікс, 2010. 544 с.
4. Трофименко О. Г. Основи програмування. Базові алгоритми : метод. вказівки для лаб. і практ. робіт / Трофименко О. Г., Прокоп Ю. В., Швайко І. Г., Буката Л. М. Ч. 1. Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2014. 108 с.
5. Трофименко О. Г. Основи програмування. Опрацювання структурованих типів : метод. вказівки для лаб. і практ. робіт / Трофименко О. Г., Прокоп Ю. В., Швайко І. Г., Буката Л. М. Ч. 2.

Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2014. 132 с.

6. Трофименко О. Г. Основи програмування. Програмне опрацювання файлів : метод. вказівки для лаб. і практ. робіт / Трофименко О. Г., Прокоп Ю. В., Швайко І. Г., Буката Л. М. Ч. 3. Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2015. 78 с.

7. Берковський В.В., Левтеров А.І., Костикова М.В., Онуфрей Ю.Є., Подоляка О.О., Попеленко А. А. Програмування в середовищі C(C++). Збірник задач. - Харків: ХНАДУ, 2016. - 224 с.

8. Вступ до програмування мовою C++. Організація обчислень : навч. посіб. / Ю. А. Белов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. - К. : Видавничо-поліграфічний центр "Київський університет", 2012. - 175 с.9.

9. Приходько, С. Б. Методичні вказівки до виконання лабораторних робіт з дисципліни "Основи програмування": у 2 ч. Ч. 1 / С. Б. Приходько, Л. М. Макарова, Т. Г. Смикодуб. – Миколаїв: НУК, 2022. – 60 с.

10. Тімонін В.О. Методичні вказівки до проведення практичних робіт з дисципліни «Алгоритмізація та програмування» - Харків: ХНАДУ, 2018. - 84 с.

Допоміжна

1. Трофименко О. Г. Створення багатомодульних програмних проектів для опрацювання даних у файлах засобами Visual C++: метод. вказівки для виконання курсової роботи з дисципліни "Основи програмування" / Трофименко О. Г., Прокоп Ю. В. Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2015. 42 с.

2. Воробйова О.Д., Глазунова Л.В. Алгоритми та структури даних. Ч. 1. Структури даних : конспект лекцій. Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2017. 48 с.

3. Воробйова О.Д., Глазунова Л.В. Алгоритми та структури даних Ч. 2. Алгоритми пошуку, стиснення даних, внутрішнього та зовнішнього сортування, алгоритми на графах : конспект лекцій. Одеса: ВЦ ОНАЗ ім. О. С. Попова, 2017. 52 с.

Інформаційні ресурси

1. <http://msdn.microsoft.com/> – Бібліотека класів платформи .NET Framework.
2. <https://classes.mst.edu/compsci1570/> - Programming C++ (Missouri S&T Computer Science) – курс лекцій з основ програмування C++ з відеоуроками (англійською мовою)
3. <https://ideone.com/> – онлайн компілятор різними мовами програмування.
4. <http://cpp.sh/> – онлайн компілятор мовою C++.
5. <http://codepad.org/> – онлайн компілятор мовами C, C++ та ін.
6. <https://www.koding.com/> – компілятор різними мовами з використанням хмарних технологій.