

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ

Декаан факультету інформаційних технологій



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Програмування

Назва дисципліни

Галузь знань – G Інженерія, виробництво та будівництво

Спеціальність – G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Рівень вищої освіти – Перший (бакалаврський)

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

Обсяг дисципліни – 15 кредитів ЄКТС, **Шифр дисципліни** – ОФП.01

Мова навчання – українська

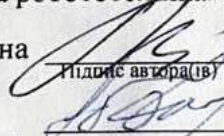
Статус дисципліни: обов'язкова (фахової підготовки)

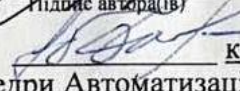
Факультет – Інформаційних технологій

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

Форма здобуття освіти	Курс	Семестр	Загальний обсяг		Кількість годин						Курсовий проєкт	Курсова робота	Форма семестрового контролю	
					Аудиторні заняття					Самостійна робота, у т.ч. ІРС			Залік	Іспит
			Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття					
Д	1	1	6	180	66	16	50			114				+
Д	1	2	4	120	50	16	34			70				+
Д	2	3	5	150	50	16	34			100				+
Разом ДФН			15	450	166	48	118			284				3

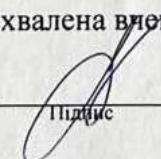
Робоча програма складена на основі освітньо-професійної програми «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» за спеціальністю G7 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

Робоча програма складена  канд. техн. наук, доцент Галина РАДЕЛЬЧУК
Підпис автора(ів) Науковий ступінь, вчене звання, ім'я, ПРІЗВИЩЕ автора(ів)

 канд. техн. наук, доцент Юрій ФОРКУН
Схвалена на засіданні кафедри Автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

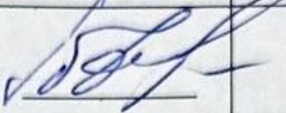
Протокол від 01.09.2025 № 1. Зав. кафедри  Людмила КОРЕЦЬКА
Підпис Ім'я, ПРІЗВИЩЕ

Робоча програма розглянута та схвалена вченою радою факультету інформаційних технологій

Голова вченої ради факультету  Тетяна ГОВОРУЩЕНКО
Підпис Ім'я, ПРІЗВИЩЕ

Хмельницький 2025

ЛИСТ ПОГОДЖЕННЯ

Посада	Назва кафедри	Підпис	Ім'я, ПРИЗВИЩЕ
Завідувач кафедри, канд. техн. наук, доц.	Автоматизації, комп'ютерно-інтегрованих технологій та робототехніки		Людмила КОРЕЦЬКА
Гарант освітньо-професійної програми, канд. техн. наук, доц.	Автоматизації, комп'ютерно-інтегрованих технологій та робототехніки		Юрій ФОРКУН

3 Пояснювальна записка

Дисципліна «Програмування» є однією з дисциплін фахової підготовки і займає провідне місце у підготовці здобувачів першого (бакалаврського) рівня вищої освіти очної (денної) (далі – денної) форми здобуття вищої освіти, які навчаються за освітньо-професійною програмою «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» у межах спеціальності G7 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

Пререквізити – вихідна.

Постреквізити – Веб-технології та бази даних (ОФП.02), Веб-технології та бази даних – курсовий проєкт (ОФП.03), Основи мехатроніки та робототехніки (ОФП.04), Робототехніка та штучний інтелект (ОФП.06), Комп'ютерна схематехніка та мікропроцесорна техніка (ОФП.07), Програмування мікропроцесорних систем керування (ОФП.08), Системний аналіз, моделювання процесів та систем (ОФП.09), Людино-машинний інтерфейс (ОФП.14).

Відповідно до освітньої програми дисципліна сприяє забезпеченню:

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 (ІК); здатність застосовувати знання у практичних ситуаціях (ЗК.01); здатність спілкуватися державною мовою як усно, так і письмово (ЗК.02); навички використання інформаційних і комунікаційних технологій (ЗК.04); здатність використовувати для вирішення професійних завдань новітні технології у галузі автоматизації, комп'ютерно-інтегрованих технологій та робототехніки, 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 (ФК.06); 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 (ФК.07); 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 та робототехніки (ФК.09); здатність здійснювати аналіз, вибір, обґрунтування та налагодження автоматизованих та інтелектуальних робототехнічних систем на основі їх технічних характеристик, принципів їх роботи, вимог до комп'ютерно-інтегрованих систем управління, а також використовувати методи та алгоритми штучного інтелекту і комп'ютерний зір в робототехніці (УК.01); здатність інтегрувати інформаційні технології при проєктуванні систем автоматизації, комп'ютерно-інтегрованих технологій та робототехніки (УК.02); здатність інтегрувати новітні технології, сучасні методи створення інтернет-ресурсів і мови програмування для розроблення алгоритмічного та програмного забезпечення систем автоматизації, комп'ютерно-інтегрованих технологій та робототехнічних систем (УК.03).

програмних результатів навчання: вміти застосовувати сучасні інформаційні технології та мати навички розробляти алгоритми та комп'ютерні програми з використанням мов високого рівня та технологій об'єктно-орієнтованого програмування, створювати бази даних та використовувати інтернет-ресурси (ПРН03); 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 (ПРН09); 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 (ПРН10); застосовувати робототехнічні системи, інтелектуальну робототехніку, обґрунтовувати їх вибір на основі аналізу їх роботи, властивостей, призначення і технічних характеристик з урахуванням вимог до комп'ютерно-інтегрованих систем управління; здійснювати налагодження робототехнічних систем, а також використовувати методи та алгоритми штучного інтелекту і комп'ютерний зір в робототехніці (ПРН15); застосовувати інтегровані знання програмно-технічних засобів та інформаційних технологій систем автоматизації, комп'ютерних систем керування та робототехніки для вирішення технічних задач спеціальності (ПРН16); 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 (ПРН17).

Мета дисципліни. Формування у здобувачів вищої освіти компетентностей, необхідних для розв'язання складних спеціалізованих задач у сфері автоматизації, комп'ютерно-інтегрованих технологій та робототехніки, зокрема, формування базових теоретичних знань та практичних навичок, спрямованих на вирішення завдань проєктування та розроблення програмного забезпечення (ПЗ) з використанням мов високого рівня і сучасних технологій програмування.

Предмет дисципліни. Теорія і практика застосування базових алгоритмічних структур, базових і похідних структур даних, засобів мов програмування високого рівня та сучасних технологій програмування при розробленні програмного забезпечення об'єктів автоматизації та робототехніки.

Завдання дисципліни. Надати здобувачам вищої освіти теоретичні знання і практичні навички з проєктування та розроблення ПЗ; сформувати в них базові компетентності, необхідні для подальшого вивчення циклу професійно-орієнтованих дисциплін.

Результати навчання. Після вивчення дисципліни студент повинен: досконало володіти професійною термінологією та основними поняттями у сфері розроблення ПЗ, вміло використовувати понятійний апарат; володіти сучасними методами, мовами програмування і технологіями розроблення алгоритмічного та програмного забезпечення об'єктів автоматизації та робототехніки; 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 високого рівня (C, C++, C#) з урахуванням особливостей базових алгоритмічних структур та сучасних технологій програмування; 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 заняття	СРС
	<i>Перший семестр</i>		
Тема 1. Загальний процес комп'ютерного розв'язування задач.	2	6	13
Тема 2. Огляд мови C та середовища програмування Microsoft Visual Studio.	4	6	25
Тема 3. Структурне програмування.	4	24	30
Тема 4. Регулярні типи даних мови C. Масиви та рядки.	6	14	46
Разом за 1-й семестр:	16	50	114
	<i>Другий семестр</i>		
Тема 5. Вказівники. Динамічний розподіл пам'яті у мові C.	4	8	15
Тема 6. Похідні типи даних мови C. Структури та об'єднання.	2	4	8
Тема 7. Процедурне програмування.	6	12	26
Тема 8. Візуальне програмування мовою C++.	2	8	10
Тема 9. Модульна організація програм.	2	2	11
Разом за 2-й семестр:	16	34	70
	<i>Третій семестр</i>		
Тема 10. Технології об'єктно-орієнтованого проєктування програмних систем.	4	4	22
Тема 11. Класи та об'єкти у мові C#. Абстрактні класи.	4	12	26
Тема 12. Концепція наслідування. Інтерфейси. Множинне наслідування.	2	8	11
Тема 13. Делегати та події. Універсальність.	6	8	41
Разом за 3-й семестр:	16	34	100

5 Програма навчальної дисципліни

5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>Перший семестр</i>	
	<i>Тема 1. Загальний процес комп'ютерного розв'язування задач.</i>	2
1	Загальний процес комп'ютерного розв'язування задач. Алгоритмізація задач. Етапи розв'язування задач на ПК. Алгоритм, його властивості та способи представлення. Базові алгоритмічні структури. Поняття програми, мови програмування, транслятора, системи програмування. Тестування та відлагодження програми. Літ.: [1] с. 12-41; [2] с. 11-18; [3] с. 117-134	2
	<i>Тема 2. Огляд мови C та середовища програмування Microsoft Visual Studio.</i>	4
2	Мова програмування C/C++. Основні конструкції. Структура програми. Середовище розроблення програм. Загальний огляд C/C++. Синтаксис мови C. Типи даних. Змінні та константи. Стандартні математичні функції. Операції та вирази. Перетворення типів. Структура C-програми. Інтегровані середовища програмування. Середовище розроблення Microsoft Visual Studio. Літ.: [1] с. 45-58; [2] с. 19-38; [3] с. 11-116	2
3	Консольне введення/виведення даних. Організація консольного введення та виведення даних у мові C. Форматоване введення/виведення даних. Функції printf() та scanf(). Введення/виведення символів та рядків. Введення/виведення у стилі C++. Локалізація консольних застосунків. Літ.: [1] с. 77-91; [2] с. 39-50; [9]	2
	<i>Тема 3. Структурне програмування.</i>	4
4	Розроблення консольних застосунків лінійної та розгалуженої структури. 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. Літ.: [1] с. 66-76, 103-110; [2] с. 51-62; [3] с. 135-158	2
5	Програмування циклічних алгоритмічних структур. Арифметичні та ітераційні цикли. Оператори циклу for, while, do...while. Вкладені цикли. Оператори break та continue. Типові приклади. Літ.: [1] с. 124-134; [2] с. 63-75; [3] с. 159-170	2
	<i>Тема 4. Регулярні типи даних мови C. Масиви та рядки.</i>	6
6	Числові масиви у мові C. Робота з одновимірними масивами. Поняття масиву у мові C. Одновимірні масиви. Оголошення та ініціалізація. Формування, опрацювання та виведення одновимірних масивів. Наслідки виходу за межі діапазону зміни індекса масиву. Типові приклади. Літ.: [1] с. 217-220; [2] с. 76-119; [3] с. 195-214	2
7	Символьні масиви у мові C (рядки). Деякі прийоми опрацювання символьних даних. Поняття символьного масиву. Оголошення та ініціалізація символьних масивів. Введення, опрацювання та виведення символьних масивів (рядків). Бібліотечні функції мови C для роботи з рядками. Типові приклади. Літ.: [1] с. 276-289; [2] с. 155-170; [3] с. 287-294	2
8	Багатовимірні масиви. Організація двовимірних та багатовимірних масивів. Оголошення та ініціалізація. Опрацювання двовимірних масивів. Реалізація векторно-матричних операцій. Опрацювання багатовимірних масивів. Масиви рядків. Типові приклади. Літ.: [1] с. 221-232; [2] с. 120-154; [3] с. 195-214	2
	Разом за 1-й семестр:	16

	Другий семестр	
	<i>Тема 5. Вказівники. Динамічний розподіл пам'яті у мові С.</i>	4
1	Вказівники. Вказівники та масиви. 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. 184-189; [3] с. 245-268	2
2	Динамічний розподіл пам'яті. Функції динамічного розподілу пам'яті та їх застосування. 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. 190-204	2
	<i>Тема 6. Похідні типи даних мови С. Структури та об'єднання.</i>	2
3	Похідні типи даних мови С. Структури та об'єднання. Способи створення похідних типів даних у мові С. Тип перерахування. Перейменовування типів (typedef). Структури (struct). Масиви структур. Вкладені структури. Об'єднання (union). Бітові поля. Літ.: [1] с. 297-305; [2] с. 243-266	2
	<i>Тема 7. Процедурне програмування.</i>	6
4	Функції у мові С. Оголошення (опис) та визначення функцій. Обмін даними між функціями. 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. 205-227; [3] с. 215-244	2
5	Функції і вказівники. Масиви як параметри функцій. Функції, які повертають значення через вказівник. Вказівники на функції. Способи передачі структур у функції, Створення функцій, які повертають структури і вказівники на структури. Рекурсивні алгоритми та функції. Літ.: [1] с. 180-188, 225-232, 302; [2] с. 228-242; [3] с. 215-244.	2
6	Файлове введення/виведення. Поняття потоку і файлу в мові 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. 267-286	2
	<i>Тема 8. Візуальне програмування мовою С++.</i>	2
7	Візуальне програмування мовою С++. Поняття Windows-форми. Властивості, методи та події класу Form. Створення проєкту Windows Forms (Visual C++). Головний файл .cpp проєкту та заголовний файл .h. Інструменти візуального розроблення форм. Елементи управління. Приклади елементів управління та їх використання. Розроблення обробників подій та допоміжних процедур і функцій. Літ.: [4] с. 7-38, 80-102	2
	<i>Тема 9. Модульна організація програм.</i>	2
8	Модульна організація програм. Поняття модуля і модульного програмування. Структура складної С-програми. Багатофайлові С-програми. Міжфайлова взаємодія. Заголовні файли (.h-файли) і файли реалізації (.c-файли). Директиви препроцесора. Стандартні бібліотеки (.lib). Модулі у мові С++. Літ.: [1] с. 201-208	2
Разом за 2-й семестр:		16

	Третій семестр	
	<i>Тема 10. Технології об'єктно-орієнтованого проектування програмних систем.</i>	4
1	Вступ до об'єктно-орієнтованого програмування. Сучасні технології та платформи проектування програмних систем. Парадигми програмування. Об'єктно-орієнтований підхід до розроблення програмних застосунків. Фундаментальні поняття об'єктно-орієнтованого програмування (ООП). Критика ООП. Літ.: [1] с. 12-20; [2] с. 11-19; [8] с. 8-32; [10]	2
2	Особливості створення застосунків у .NET Framework та особливості мови програмування C#. Платформа .NET. Базові поняття .NET Framework. Основні конструкції та особливості мови C#. Сучасні інтегровані середовища об'єктно-орієнтованого програмування мовою C#. Створення застосунків за допомогою .NET Framework та мови програмування C#. Літ.: [7] с. 4-268; [8] с. 19-32	2
	<i>Тема 11. Класи та об'єкти у мові C#. Абстрактні класи.</i>	4
3	Класи та об'єкти C#. Основні поняття. Оголошення та структура класу. Дані, властивості та методи класу. Декларації private, protected, public. Ініціалізація об'єктів класу. Конструктори та деструктори, їх призначення, оголошення та варіанти викликів. Звичайні, константні і статичні дані та методи, особливості їх оголошення. Отримання детальної інформації про клас. Літ.: [7] с. 69-114, 131-135, 197-226; [8] с. 19-68	2
4	Абстрактні класи. Структури та перерахування. Поняття про абстрактні класи. Розгорнуті й вказівкові типи. Класи й структури. Структури. Вбудовані структури. Перерахування та їх особливості. Літ.: [7] с. 136-139; [8] с. 88-93	2
	<i>Тема 12. Концепція наслідування. Множинне наслідування.</i>	2
5	Концепція наслідування. Множинне наслідування. Поняття про наслідування (успадкування) та ієрархія класів. Реалізація успадкування в C#. Додавання полів нащадкам. Конструктори батьків і нащадків. Додавання методів і зміна методів батька. Статичний контроль типів та динамічне зв'язування. Класи без нащадків. Множинне наслідування. Успадкування від загального батьківського класу. Проблеми множинного наслідування. Колізія імен. Літ.: [7] с. 116-135; [8] с. 69-78	2
	<i>Тема 13. Інтерфейси, делегати та події. Універсальність.</i>	6
6	Інтерфейси та делегати. Означення інтерфейсу. Способи реалізації інтерфейсу. Перетворення до класу інтерфейсу. Вбудовані інтерфейси. Означення делегата. Функції вищих порядків. Операції над делегатами. Клас Delegate та його основні методи і властивості. Приклади. Літ.: [7] с. 168-196; [8] с. 94-111, 259-269	2
7	Події. Загальні поняття. Події для багатоадресної передачі. Порівняння методів екземплярів зі статичними методами, що використовуються в якості обробників подій. Використання подійних засобів доступу. Рекомендації по обробленню подій у середовищі .NET Framework. Використання подій. Приклади. Літ.: [8] с. 259-269	2
8	Універсальність. Класи з родовими параметрами. Спадкування та універсальність. Синтаксис універсального класу. Клас з універсальними методами. Два основних механізми об'єктної технології. Стек. Від абстрактного (універсального) класу до конкретних з версіями. Обмежена універсальність. Синтаксис обмежень. Список з можливістю пошуку елементів по ключу. Родове породження класу. Універсальність і спеціальні випадки класів. Універсальні структури, інтерфейси та делегати. .Net Framework та універсальність. Літ.: [8] с. 79-90	2
Разом за 3-й семестр:		16

5.2 Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
<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 (з однократними циклами) Літ.: [5] с. 72-102; [6]	6
6	Розроблення С-програм циклічної структури (із вкладеними циклами) Літ.: [5] с. 72-102; [6]	6
7	Опрацювання одновимірних масивів та рядків Літ.: [5] с. 122-140, 201-228; [6]	6
8	Робота з багатовимірними масивами Літ.: [5] с. 141-163; [6]	6
9	Підсумкове заняття	2
Разом за 1-й семестр:		50
<i>Другий семестр</i>		
1	Робота з вказівниками. Опрацювання динамічних одновимірних масивів Літ.: [5] с. 164-184; [6]	4
2	Робота з вказівниками. Опрацювання динамічних двовимірних масивів Літ.: [5] с. 164-184; [6]	4
3	Опрацювання структур Літ.: [5] с. 252-260; [6]	4
4	Розроблення С-програм з використанням функцій користувача: основи роботи Літ.: [5] с. 105-121, 185-200; [6]	4
5	Розроблення С-програм з використанням функцій користувача: додаткові можливості Літ.: [5] с. 105-121, 185-200; [6]	4
6	Робота з файлами даних Літ.: [5] с. 229-250, 276-298; [6]	4
7	Створення проекту застосунку Windows Forms засобами мови C++: основи роботи Літ.: [4] с. 7-30, 88-102	4
8	Створення проекту застосунку Windows Forms засобами мови C++: додаткові можливості Літ.: [4] с. 7-30, 88-102	4
9	Підсумкове заняття	2
Разом за 2-й семестр:		34
<i>Третій семестр</i>		
1	Вступ до C#. Створення консольних C#-застосунків Літ.: [7] с. 4-68, 154-167	4
2	Використання класів C# для роботи з середовищем та файловою системою ОС Windows Літ.: [7] с. 197-226; [8] с. 33-68	4
3	Створення інтерфейсу користувача. Робота об'єктами, подіями та елементами керування. Літ.: [7] с. 197-226; [8] с. 158-232, 270-300	4
4	Робота з класами та об'єктами. Інкапсуляція Літ.: [7] с. 69-135; [8] с. 33-68	4

5	Структури та перерахування Літ.: [7] с. 136-139; [8] с.79-93	4
6	Відносини між класами. Наслідування. Клієнти і нащадки. Літ.: [7] с. 116-135; [8] с. 69-78	4
7	Інтерфейси. Множинне наслідування Літ.: [7] с. 168-196; [8] с. 94-111	4
8	Делегати. Класи з подіями. Літ.: [8] с. 259-269	4
9	Підсумкове заняття	2
Разом за 3-й семестр:		34

5.3 Зміст самостійної (у т.ч. індивідуальної) роботи здобувача вищої освіти

Самостійна робота студентів денної форми здобуття освіти полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації, підготовці до виконання і захисту лабораторних робіт, тематичне тестування. Окрім цього до послуг студентів сторінка навчальної дисципліни у Модульному середовищі для навчання, де розміщені Робоча програма дисципліни та необхідні документи з її навчально-методичного забезпечення.

Номер тижня	Вид самостійної роботи	Кількість годин
<i>Перший семестр</i>		
1	Опрацювання теоретичного матеріалу з Т1, підготовка до виконання ЛР1.	6
2	Опрацювання теоретичного матеріалу з Т1, підготовка до захисту ЛР1, підготовка до виконання ЛР2.	7
3	Опрацювання теоретичного матеріалу з Т2, підготовка до захисту ЛР2.	6
4	Опрацювання теоретичного матеріалу з Т2, підготовка до захисту ЛР2, підготовка до виконання ЛР3.	7
5	Опрацювання теоретичного матеріалу з Т2, підготовка до захисту ЛР3, підготовка до виконання ЛР4.	6
6	Опрацювання теоретичного матеріалу з Т2, підготовка до захисту ЛР3, підготовка до виконання ЛР4.	6
7	Опрацювання теоретичного матеріалу з Т3. Підготовка до захисту ЛР4. Підготовка до виконання ЛР5, підготовка до ТК1.	8
8	Опрацювання теоретичного матеріалу з Т3. Підготовка до захисту ЛР4. Підготовка до виконання ЛР5, підготовка до ТК1.	8
9	Опрацювання теоретичного матеріалу з Т3, підготовка до захисту ЛР5, підготовка до виконання ЛР6.	7
10	Опрацювання теоретичного матеріалу з Т3, підготовка до захисту ЛР5, підготовка до виконання ЛР6.	7
11	Опрацювання теоретичного матеріалу з Т4, підготовка до захисту ЛР6, підготовка до виконання ЛР7.	6
12	Опрацювання теоретичного матеріалу з Т4, підготовка до захисту ЛР6, підготовка до виконання ЛР7.	6
13	Опрацювання теоретичного матеріалу з Т4. Підготовка до захисту ЛР7. Підготовка до виконання ЛР8.	6
14	Опрацювання теоретичного матеріалу з Т4. Підготовка до захисту ЛР7. Підготовка до виконання ЛР8. Підготовка до ТК2.	8
15	Опрацювання теоретичного матеріалу з Т4, підготовка до захисту ЛР8. Підготовка до ТК2.	8
16	Опрацювання теоретичного матеріалу з Т4, підготовка до захисту ЛР8.	6
17	Опрацювання теоретичного матеріалу.	6
Разом за 1-й семестр:		114

13	Опрацювання теоретичного матеріалу з Т13. Підготовка до захисту ЛР7. Підготовка до виконання ЛР8.	5
14	Опрацювання теоретичного матеріалу з Т13. Підготовка до захисту ЛР7. Підготовка до виконання ЛР8. Підготовка до ТК2 (Т 12-13).	7
15	Опрацювання теоретичного матеріалу з Т13, підготовка до захисту ЛР8. Підготовка до ТК2 (Т 12-13).	7
16	Опрацювання теоретичного матеріалу з Т13, підготовка до захисту ЛР8.	5
17	Опрацювання теоретичного матеріалу.	5
Разом за 3-й семестр:		100

Примітка. Т – тема навчальної дисципліни, ТК – тестовий контроль, ЛР – лабораторна робота.

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

6 Технології та методи навчання

Процес навчання з дисципліни ґрунтується на застосуванні традиційних та сучасних технологій і методів навчання, зокрема: лекції (з використанням методів візуалізації, проблемного й інтерактивного навчання, мотиваційних прийомів, інформаційно-комунікаційних технологій); лабораторні заняття (з використанням методів інформаційних технологій та сучасних інтегрованих середовищ програмування, майстер-класів, практикумів тощо); самостійна робота (опрацювання теоретичного матеріалу, підготовка до виконання та захисту лабораторних робіт, поточного та підсумкового контролю) з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

7 Методи контролю

Поточний контроль здійснюється під час аудиторних лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком освітнього процесу, у т. ч. з використанням Модульного середовища для навчання. При цьому використовуються такі методи поточного контролю:

- усне опитування перед допуском до лабораторного заняття;
- оцінювання результатів захисту лабораторних робіт;
- тестовий контроль засвоєння теоретичного та практичного матеріалу (за темами).

При виведенні підсумкової семестрової оцінки враховуються результати як поточного контролю, так і підсумкового контролю, який проводиться з усього матеріалу дисципліни за білетами, попередньо розробленими і затвердженими на засіданні кафедри. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи суму балів нижчу за 60 відсотків від максимального балу, **не допускається** до семестрового контролю, поки не виконає обсяг роботи, передбачений Робочою програмою. Здобувач вищої освіти, який набрав позитивний середньозважений бал (60 відсотків і більше від максимального балу) з усіх видів поточного контролю і не склав іспит, вважається таким, який **має** академічну заборгованість. Ліквідація академічної заборгованості із семестрового контролю здійснюється у період екзаменаційної сесії або за графіком, встановленим деканатом відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ».

8 Політика дисципліни

Політика навчальної дисципліни загалом визначається системою вимог до здобувача вищої освіти, що передбачені чинними положеннями Університету про організацію і навчально-методичне забезпечення освітнього процесу. Зокрема, проходження інструктажу з техніки безпеки та відвідування занять з дисципліни є обов'язковим. За об'єктивних причин (підтверджених документально) теоретичне навчання за погодженням із лектором може відбуватись в онлайн-режимі. Успішне опанування дисципліни і формування фахових компетентностей та програмних результатів навчання передбачає необхідність підготовки до лабораторного заняття: вивчення теоретичного матеріалу з теми роботи, попередню підготовку протоколу роботи, підготовку до усного опитування для допуску до заняття (наведені у Методичних рекомендаціях до лабораторних занять); активно працювати на занятті; якісно підготувати звіт (протокол роботи відповідно до теми); захистити результати виконаної роботи; брати участь у дискусіях щодо прийнятих конструктивних рішень при виконанні здобувачами лабораторних робіт тощо.

Здобувачі вищої освіти мають дотримуватися встановлених термінів виконання всіх видів навчальної роботи відповідно до робочої програми навчальної дисципліни. Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. Пропущене лабораторне заняття студент зобов'язаний відпрацювати у встановлений викладачем термін, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється за результатами опитування під час лабораторних занять та тематичного тестування.

Здобувач вищої освіти, виконуючи самостійну або індивідуальну роботу з дисципліни, має дотримуватися політики доброчесності (заборонені списування, плагіат, у т. ч. з використанням мобільних девайсів. У разі виявлення порушення політики академічної доброчесності в будь-яких видах навчальної роботи здобувач вищої освіти отримує незадовільну оцінку і має повторно виконати завдання з відповідної теми (виду роботи), що передбачені робочою програмою. Будь-які форми порушення академічної доброчесності **не допускаються**.

У межах вивчення навчальної дисципліни здобувачам вищої освіти передбачено визнання та зарахування результатів навчання, набутих шляхом неформальної освіти, що розміщені на доступних платформах, які сприяють формуванню компетентностей та поглибленню результатів навчання, визначених робочою програмою дисципліни, або забезпечують вивчення відповідної теми та/або виду робіт з програми навчальної дисципліни (детальніше – у Положенні про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ).

9 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». При поточному оцінюванні виконаної здобувачем роботи з кожної структурної одиниці і отриманих ним результатів викладач виставляє йому певну кількість балів із встановлених Робочою програмою для цього виду роботи. При цьому кожна структурна одиниця навчальної роботи може бути зарахована, якщо здобувач набрав не менше 60 відсотків (мінімальний рівень для позитивної оцінки) від максимально можливої суми балів, призначеної структурній одиниці.

При оцінюванні результатів навчання здобувачів вищої освіти з будь-якого виду навчальної роботи (структурної одиниці) рекомендується використовувати наведені нижче узагальнені критерії:

Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти

Оцінка та рівень досягнення здобувачем запланованих ПРН та сформованих компетентностей	Узагальнений зміст критерію оцінювання
Відмінно (високий)	Здобувач вищої освіти 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 та обґрунтовувати свої судження. Відмінна оцінка передбачає логічний виклад відповіді мовою викладання (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними приладами та інструментами, прикладними програмами. Здобувач не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки, демонструє практичні навички з вирішення фахових завдань. При відповіді допустив дві-три несуттєві похибки .
Добре (середній)	Здобувач вищої освіти 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 та формі відповіді можуть мати місце окремі неточності, нечіткі формулювання правил, закономірностей тощо. Відповідь здобувача вищої освіти будується на основі самостійного мислення. Здобувач вищої освіти у відповіді допустив дві-три несуттєві помилки .
Задовільно (достатній)	Здобувач вищої освіти 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 здобувача вищої освіти будується на рівні репродуктивного мислення, здобувач вищої освіти має слабкі знання структури навчальної дисципліни, 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.
Незадовільно (недостатній)	Здобувач вищої освіти 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 здобувачеві вищої освіти, який не може продовжити навчання без додаткової роботи з вивчення навчальної дисципліни.

Структурування дисципліни за видами навчальної роботи і оцінювання результатів навчання студентів денної форми здобуття освіти у семестрі

Аудиторна робота								Контрольні заходи		Семестровий контроль	
Перший семестр											
Лабораторні роботи №:								Тестовий контроль:		Іспит	Разом балів
1	2	3	4	5	6	7	8	Т 1-2	Т 3-4		
Кількість балів за вид навчальної роботи (мінімум-максимум)											60-100
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	6-10	6-10	24-40	
24-40								12-20		24-40	
Другий семестр											
Лабораторні роботи №:								Тестовий контроль:		Іспит	Разом балів
1	2	3	4	5	6	7	8	Т 5-6	Т 7-9		
Кількість балів за вид навчальної роботи (мінімум-максимум)											60-100
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	6-10	6-10	24-40	
24-40								12-20		24-40	
Третій семестр											
Лабораторні роботи №:								Тестовий контроль:		Іспит	Разом балів
1	2	3	4	5	6	7	8	Т 10-11	Т 12-13		
Кількість балів за вид навчальної роботи (мінімум-максимум)											60-100
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	6-10	6-10	24-40	
24-40								12-20		24-40	

Примітка. За набрану з будь-якого виду навчальної роботи з дисципліни кількість балів, нижче встановленого мінімуму, здобувач отримує незадовільну оцінку і має її перездати у встановлений викладачем (деканом) термін. Інституційна оцінка встановлюється відповідно до таблиці «Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС».

Оцінювання результатів захисту лабораторної роботи

Виконана й оформлена відповідно до встановлених Методичними рекомендаціями вимог лабораторна робота комплексно оцінюється викладачем при її захисті з урахуванням таких критеріїв: самостійність та правильність виконання; повнота відповіді та знання з відповідної теми лабораторної роботи; дотримання вимог при оформленні звіту з лабораторної роботи.

Результат виконання та захисту здобувачем вищої освіти кожної лабораторної роботи оцінюється відповідно до таблиці Критеріїв оцінювання навчальних досягнень здобувача вищої освіти (мінімальний позитивний бал – 3 бали, максимальний – 5 балів).

У випадку виявлення здобувачем рівня знань, нижчого ніж 60 відсотків від максимального балу, встановленого Робочою програмою для кожної структурної одиниці, лабораторна робота йому **не зараховується** і для її захисту він має детальніше опрацювати матеріал з теми роботи, методiku її виконання, виправити грубі помилки та повторно вийти на її захист у призначений викладачем час.

Оцінювання результатів тестового контролю

Кожний з тестів, передбачених Робочою програмою, складається із 25 тестових завдань, кожне з яких є рівнозначним.

Відповідно до таблиці структурування видів робіт за тестовий контроль здобувач, залежно від кількості правильних відповідей, може отримати від 6 до 10 балів.

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

Кількість правильних відповідей	1-13	14-16	17-18	19-20	21-22	23-25
Відсоток правильних відповідей	0-59	60-65	66-72	73-82	83-89	90-100
Кількість балів	-	6	7	8	9	10

На тестування відводиться 30-40 хвилин. Студент проходить тестування в онлайн-режимі у Модульному середовищі для навчання. Також студент може проходити тестування письмово, записуючи правильні відповіді у талоні відповідей. При отриманні негативної оцінки тест слід перездати до терміну **наступного** контролю.

Оцінювання результатів підсумкового семестрового контролю (іспит)

Освітня програма передбачає підсумковий семестровий контроль з дисципліни у формі іспиту (у кожному семестрі), завданням якого є системне й об'єктивне оцінювання як теоретичної, так і практичної підготовки здобувача з навчальної дисципліни. Складання іспиту відбувається за попередньо розробленими і затвердженими на засіданні кафедри білетами. Відповідно до цього в екзаменаційному білеті пропонується поєднання питань як теоретичного (в т. ч. у тестовій формі), так і практичного характеру.

Таблиця – Оцінювання результатів підсумкового семестрового контролю здобувачів денної форми навчання (40 балів для підсумкового контролю)

Види завдань	Для кожного окремого виду завдань		
	Мінімальний (достатній) бал (задовільно)	Потенційні позитивні бали* (середній бал) (добре)	Максимальний (високий) бал (відмінно)
Теоретичне питання № 1	6	8	10
Теоретичне питання № 2	6	8	10
Практичне завдання	12	15	20
Разом:	24		40

Примітка. *Позитивний бал за іспит, відмінний від мінімального (24 бали) та максимального (40 балів), знаходиться в межах 25-39 балів та розраховується як сума балів за усі структурні елементи (завдання) іспиту.

Для кожного окремого виду завдань підсумкового семестрового контролю застосовуються критерії оцінювання навчальних досягнень здобувача вищої освіти, наведені у **Таблиці – Критерії оцінювання навчальних досягнень здобувача вищої освіти**.

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС визначається в автоматизованому режимі після внесення викладачем результатів оцінювання у балах з усіх видів навчальної роботи до електронного журналу. Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені нижче у таблиці «Співвідношення».

Семестровий іспит виставляється, якщо загальна сума балів, яку набрав студент з дисципліни за результатами поточного та семестрового контролю, знаходиться у межах від 60 до 100 балів. При цьому за інституційною шкалою ставиться оцінка «відмінно/добре/задовільно», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом сумі балів відповідно до таблиці «Співвідношення».

Таблиця – Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Рейтингова шкала балів	Інституційна оцінка (рівень досягнення здобувачем вищої освіти запланованих результатів навчання з навчальної дисципліни)
		Іспит
A	90-100	Відмінно/Excellent – високий рівень досягнення запланованих результатів навчання з навчальної дисципліни, що свідчить про безумовну готовність здобувача до подальшого навчання та/або професійної діяльності за фахом
B	83-89	Добре/Good – середній (максимально достатній) рівень досягнення запланованих результатів навчання з навчальної дисципліни та готовності до подальшого навчання та/або професійної діяльності за фахом
C	73-82	
D	66-72	Задовільно/Satisfactory – наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
E	60-65	
FX	40-59	Незадовільно/Fail – низка запланованих результатів навчання з навчальної дисципліни відсутня; рівень набутих результатів навчання є недостатнім для подальшого навчання та/або професійної діяльності за фахом
F	0-39	Незадовільно/Fail – Результати навчання відсутні

10 Питання для самоконтролю результатів навчання

Перший семестр

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

Другий семестр

1. Змінні, адреси та вказівники. Оголошення та ініціалізація вказівників. Який сенс має значення вказівника NULL?
2. Основні операції над вказівниками.
3. Вказівники та одновимірні масиви.
4. Вказівники та рядки.
5. Вказівники та багатовимірні масиви.
6. Масиви вказівників, оголошення та ініціалізація. Багаторівнева адресація.
7. Способи виділення пам'яті для об'єктів у мові С. Динамічний розподіл пам'яті.
8. Функції динамічного розподілу пам'яті у мові С. Яка різниця у дії функцій **malloc()** і **calloc()**?
9. Моделювання одновимірних масивів з використанням динамічної пам'яті.
10. Динамічне виділення пам'яті для багатовимірних масивів (1-й спосіб).
11. Динамічне виділення пам'яті для багатовимірних масивів (2-й спосіб).
12. Вільні масиви. Особливості виділення динамічної пам'яті під вільний масив.
13. Охарактеризуйте тип даних «перерахування».
14. Поняття структури у мові С. Оголошення та ініціалізація. У чому полягає відмінність структури від масиву?
15. Операції зі структурами та з елементами структур.
16. Масиви структур та елементи-масиви.
17. Вкладені структури.
18. Об'єднання. Оголошення та ініціалізація.
19. Функції у мові С. Оголошення (опис) та визначення функції.
20. Параметри функцій (глобальні, локальні, формальні та фактичні). Виклик функції.
21. Загальні правила організації функцій у мові С.
22. Способи передачі даних у функцію. Оператор **return**.
23. Функції зі змінним числом параметрів.
24. Одновимірні масиви як параметри функцій. Чи може функція повернути масив?
25. Двовимірні масиви як параметри функцій.
26. Функції, які повертають значення через вказівник.
27. Опрацювання структур у функціях.
28. Рекурсивні алгоритми та функції.
29. Потоки і файли у мові С. Стандартні потоки. Текстові та бінарні файли.
30. Файлова система ANSI C. Загальна характеристика.
31. Вказівник на файл. Відкриття та закриття файла. Як у мові С кодується ознака кінця файла?
32. Запис символів у файл. Функції **getc()** і **fgetc()**.
33. Читання символів з файла. Функції **putc()** і **fputc()**.
34. Запис рядків у файл. Функції **getc()** і **fgets()**.
35. Читання рядків з файла. Функції **fputc()** та **fputs()**.
36. Форматований запис/читання у файл/з файла. Функції **fprintf()** і **fscanf()**.
37. Синтаксис та призначення функцій **feof()** та **rewind()** при роботі з файлами.
38. Читання/запис блоків даних. Функції **fread()** і **fwrite()**.
39. Довільний (прямий) доступ до файлів. Функція **fseek()**.
40. Охарактеризуйте суть модульного програмування.
41. Призначення препроцесора мови С. Основні директиви препроцесора.
42. Багатофайлові програми.
43. Заголовні файли (h-файли) та їх основні елементи.
44. Технологія візуального програмування у мові C++. Форми.
45. Інструменти візуального розроблення застосунків у мові C++.

Третій семестр

1. Сучасні технології та платформи проєктування програмних систем.
2. Парадигми програмування.
3. Об'єктно-орієнтований підхід до програмування.
4. Фундаментальні поняття ООП.
5. Базові поняття .Net Framework.

6. Основні поняття класу.
7. Оголошення та структура класу.
8. Дані, властивості та методи класу.
9. Декларації `private`, `protected`, `public`.
10. Ініціалізація об'єктів класу.
11. Конструктори та деструктори, їх призначення, оголошення та варіанти викликів.
12. Звичайні, констатні і статичні дані та методи, особливості їх оголошення.
13. Одержання детальної інформації про клас.
14. Абстрактні класи.
15. Розгорнуті й посилальні типи.
16. Класи й структури.
17. Структури.
18. Вбудовані структури.
19. Перерахування.
20. Відношення між класами.
21. Відношення "є" і "має".
22. Наслідування (успадкування).
23. Механізм наслідування (успадкування).
24. Позитивні аспекти успадкування та їх практичні можливості.
25. Реалізація успадкування у мові C#.
26. Додавання полів нащадком.
27. Конструктори батьків і нащадків.
28. Додавання методів та зміна методів батька.
29. Статичний контроль типів та динамічне зв'язування.
30. Класи без нащадків.
31. Множинне наслідування та проблеми множинного наслідування.
32. Колізія імен.
33. Успадкування від загального батьківського класу.
34. Означення інтерфейсу.
35. Два способи реалізації інтерфейсу.
36. Перетворення до класу інтерфейсу.
37. Вбудовані інтерфейси.
38. Означення делегата.
39. Функції вищих порядків.
40. Операції над делегатами.
41. Клас `Delegate`.
42. Основні методи й властивості класу `Delegate`.
43. Загальні поняття події.
44. Події для багатоадресної передачі.
45. Порівняння методів екземплярів зі статичними методами, що використовуються в якості обробника подій.
46. Використання подійних засобів доступу.
47. Рекомендації по обробці подій у середовищі .NET Framework.
48. Використання вбудованого делегата `EventHandler`.
49. Використання подій.
50. Наслідування і універсальність.
51. Синтаксис універсального класу.
52. Клас з універсальними методами.
53. Два основні механізми об'єктної технології.
54. Стек. Реалізація стека масивом.
55. Обмежена універсальність.
56. Синтаксис обмежень.
57. Список з можливістю пошуку елементів по ключу. Наслідування і вбудовування. Сумісне використання.
58. Пропозиція `using`.
59. Універсальність та спеціальні випадки класів. .Net Framework і універсальність.
60. Атрибути.

61. Перерахування і атрибут Flags.
62. Клас Attribute.
63. Статичні методи: GetCustomAttribute, GetCustomAttributes, IsDefined.
64. Атрибутні класи.
65. Атрибутний клас AttributeUsageAttribute.
66. Пов'язання атрибуту з програмним елементом.
67. Вбудовані атрибути.
68. Атрибут Serializable.
69. Атрибутний клас ConditionalAttribute.
70. Власні атрибутні класи.

11 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.

12 Матеріально-технічне та програмне забезпечення дисципліни

Інформаційна та комп'ютерна підтримка: комп'ютер, планшет, смартфон, проектор.

Програмне забезпечення: програми Microsoft Office (або аналогічні), інтегроване середовище програмування Microsoft Visual Studio (або аналоги), доступ до мережі Інтернет.

13 Рекомендована література

Основна

1. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування: навч. посіб. Львів : «Новий Світ-2000», 2020. Ч. 1. 337 с.
2. Фратавчан В. Г., Фратавчан Т. М., Лазорик В. В. Алгоритмізація та програмування: навч. посіб. Чернівці: ЧНУ, 2022. 286 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. 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.
7. Коноваленко І. В., Марущак П. О. Платформа .NET та мова програмування C# 8.0: навч. посіб. Тернопіль: ФОП Паляниця В. А., 2020. 320 с.
8. Об'єктно-орієнтоване програмування: електрон. метод. посіб. / уклад.: А. Л. Рачинська, О. А. Недева, К. С. Палій. Одеса: Одес. нац. ун-т ім. І. І. Мечникова, 2024. 338 с.
9. Радельчук Г. І. Методи вирішення проблем роботи з кирилицею у консольних C-програмах. *Вісник Хмельницького національного університету. Серія «Технічні науки»*. 2018. № 1 (257). С. 54-60.
10. Порівняння об'єктно-орієнтованої та функційної парадигм програмування у проектуванні програмного забезпечення / А. І. Коваль, О. М. Яшина, Г. І. Радельчук, Ю. В. Форкун. *Вісник Хмельницького національного університету. Серія «Технічні науки»*. 2021. № 3 (297). С. 34-38.

Додаткова

1. Seacord R. Effective C: An Introduction to Professional C Programming. Programming Paperback. San Francisco, California, No Starch Press. 2020. 272 p.
2. C Programming Tutorial. URL: <https://www.geeksforgeeks.org/c-programming-language/>. Last Updated: 18 Sep, 2025.
3. Brian H. Beej's Guide to C Programming. URL: https://beej.us/guide/bgc/pdf/bgc_usl_c_1.pdf
4. Методи розробки алгоритмів. Програмування мовою C++. URL: <https://lnuittutor.github.io/>
5. Уроки програмування на C++. URL: <https://acode.com.ua/uroki-po-cpp/>
6. Мова програмування C++. URL: <http://cpp.dp.ua/>
7. C++. Підручник. URL: <https://w3schoolsua.github.io/cpp/index.html#gsc.tab=0>

8. Прокопенко О. В., Попов М. О., Чумак Г. Л. Мова програмування C/C++. Практикум: навч. посіб. К.: Київський національний університет імені Тараса Шевченка, 2024. 375 с.
9. Цимбал О. М., Бронніков А. І Технології програмування: C++/CLI в Microsoft Visual Studio. Харків: Видавництво Іванченка І. С., 2025. 275 с.
10. Стандарт C11. URL: <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1570.pdf>
11. Introduction to C#. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/tutorials/>. Last Updated: 05.02.2025.
12. Іванов Л. В. Об'єктно-орієнтоване програмування. URL: http://iwanoff.inf.ua/oop_ua/index.html
13. Learning C# by Example. URL: <https://www.finchher.org/tips/Languages/csharp.shtml>.
14. Mark J. Price. C# 10 and .NET 6 – Modern Cross-Platform Development: Build apps, websites, and services with ASP.NET Core 6, Blazor, and EF Core 6 using Visual Studio 2022 and Visual Studio Code. URL: <https://www.amazon.com/10-NET-Cross-Platform-Development-websites/dp/1801077363>
15. Troelsen A., Japikse P. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming. Chambersburg, PA, USA, 2022. 1680 p. URL: <https://dl.ebooksworld.ir/books/Pro.CSharp.10.with.NET.6.-Andrew.Troelsen.Phil.Japikse.Apress.9781484278680.EBooksWorld.ir.pdf>
16. C#. Підручник. URL: <https://w3schoolsua.github.io/cs/index.html>

14 Інформаційні ресурси

1. Модульне середовище для навчання. URL : <https://msn.khmnu.edu.ua/course/view.php?id=9340>
2. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
3. Інституційний репозитарій ХНУ. URL : <https://elar.khmnu.edu.ua/home>

ПРОГРАМУВАННЯ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	Перший – третій
Кількість призначених кредитів ЄКТС	15,0
Форми здобуття освіти, для яких викладається дисципліна	Очна (денна)

Результати навчання. Після вивчення дисципліни студент повинен: досконало володіти професійною термінологією та основними поняттями у сфері розроблення ПЗ, вміло використовувати понятійний апарат; володіти сучасними методами, мовами програмування і технологіями розроблення алгоритмічного та програмного забезпечення об'єктів автоматизації та робототехніки; 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 високого рівня (C, C++, C#) з урахуванням особливостей базових алгоритмічних структур та сучасних технологій програмування; 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. Візуальне програмування мовою C++. Модульна організація програм. Об'єктно-орієнтований підхід до розроблення програмного забезпечення. Основні принципи та концепції об'єктно-орієнтованого програмування. Розроблення консольних та Windows-застосунків мовою C#.

Пререквізити: вихідна.

Постреквізити: Веб-технології та бази даних (ОФП.02), Веб-технології та бази даних – курсовий проєкт (ОФП.03), Основи мехатроніки та робототехніки (ОФП.04), Робототехніка та штучний інтелект (ОФП.06), Комп'ютерна схематехніка та мікропроцесорна техніка (ОФП.07), Програмування мікропроцесорних систем керування (ОФП.08), Системний аналіз, моделювання процесів та систем (ОФП.09), Людино-машинний інтерфейс (ОФП.14).

Запланована навчальна діяльність: Мінімальний обсяг навчальних занять в одному кредиті ЄКТС навчальної дисципліни для першого (бакалаврського) рівня вищої освіти за денною формою здобуття освіти становить 10 годин на 1 кредит ЄКТС.

Форми (методи) навчання: лекції (з використанням методів візуалізації, проблемного й інтерактивного навчання, мотиваційних прийомів, інформаційно-комунікаційних технологій); лабораторні заняття (з використанням методів інформаційних технологій та сучасних інтегрованих середовищ програмування, майстер-класів, практикумів тощо); самостійна робота (опрацювання теоретичного матеріалу, підготовка до виконання та захисту лабораторних робіт, поточного та підсумкового контролів) з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

Форми оцінювання результатів навчання: усне опитування перед допуском до лабораторного заняття; оцінювання результатів захисту лабораторних робіт; тестовий контроль засвоєння теоретичного та практичного матеріалу.

Вид семестрового контролю: іспит – 1, 2, 3 семестр.

Навчальні ресурси:

1. Ришкорець Ю. В., Висоцька В. А. Алгоритмізація та програмування: навч. посіб. Львів : «Новий Світ-2000», 2020. Ч. 1. 337 с.
2. Фратавчан В. Г., Фратавчан Т. М., Лазорик В. В. Алгоритмізація та програмування: навч. посіб. Чернівці: ЧНУ, 2022. 286 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. Трофименко О. Г., Прокоп Ю. В., Задерейко О. В. Алгоритмізація та програмування : навч.-метод. посіб. Одеса : Фенікс, 2020. 310 с.
6. Коноваленко І. В., Марущак П. О. Платформа .NET та мова програмування C# 8.0: навч. посіб. Тернопіль: ФОП Паляниця В. А., 2020. 320 с.
7. Об'єктно-орієнтоване програмування: електрон. метод. посіб. / уклад.: А. Л. Рачинська, О. А. Недева, К. С. Палій. Одеса: Одес. нац. ун-т ім. І. І. Мечникова, 2024. 338 с.
8. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khmnu.edu.ua/course/view.php?id=9340>
9. Електронна бібліотека ХНУ. Доступ до ресурсу: <http://library.khmnu.edu.ua/>

Викладачі: канд. техн. наук, доцент Радельчук Г., канд. техн. наук, доцент Форкун Ю.