

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ



Декан факультету _____ ФІТ

Тетяна ГОВОРУЩЕНКО
Ім'я, ПРІЗВИЩЕ

2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Об'єктно-орієнтоване програмування

Назва дисципліни

Галузь знань – F Інформаційні технології

Спеціальність – F7 Комп'ютерна інженерія

Рівень вищої освіти – Перший (бакалаврський)

Освітньо-професійна програма – Комп'ютерна інженерія та програмування (зі скороченим терміном навчання)

Обсяг дисципліни – 8.0 кредитів ЄКТС, **Шифр дисципліни** – ОФП.14

Мова навчання – українська

Статус дисципліни: обов'язкова (фахової підготовки)

Факультет – Інформаційних технологій

Кафедра – Комп'ютерної інженерії та інформаційних систем

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	8.0	240	82	16	32	34		158				+
Разом ДФН														1

Робоча програма складена на основі освітньо-професійної програми «Комп'ютерна інженерія та програмування» за спеціальністю 123 «Комп'ютерна інженерія»

Робоча програма складена _____ д-р. техн. наук, професор **Сергій ЛИСЕНКО**
Підпис автора(ів) Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ автора(ів)

Схвалена на засіданні кафедри Комп'ютерної інженерії та інформаційних систем

Протокол від 18.08 2025 № 1. Зав. кафедри _____ **Ольга ПАВЛОВА**
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	Ініціали, прізвище
Завідувач кафедри, ДФ, доц.	Комп'ютерної інженерії та інформаційних систем		Ольга ПАВЛОВА
Гарант освітньо-професійної програми, канд. техн. наук, доц.	Комп'ютерної інженерії та інформаційних систем		Андрій НІЧЕПОРУК

3. Пояснювальна записка

Дисципліна «Об'єктно-орієнтоване програмування» є однією із 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 «Комп'ютерна інженерія та програмування» в межах спеціальності F7 «Комп'ютерна інженерія».

Пререквізити – Програмування (ОФП.01), Програмування (КП) (ОФП.02)

Постреквізити – Бази даних (ОФП.04), Кваліфікаційна робота (ОФП.23).

Відповідно до освітньої програми дисципліна сприяє забезпеченню:

компетентностей: здатність розв'язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп'ютерній галузі або навчання, що передбачає застосування теорій та методів комп'ютерної інженерії і характеризується комплексністю та невизначеністю умов (ІК); здатність вчитися і оволодівати сучасними знаннями (ЗК02); здатність застосовувати знання у практичних ситуаціях (ЗК03); здатність спілкуватися державною мовою як усно, так і письмово (ЗК04); вміння виявляти, ставити та вирішувати проблеми (ЗК07); здатність застосовувати законодавчу та нормативно-правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії (ФК01); здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення (ФК02); здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж (ФК03); здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів (ФК11); здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення (ФК15); здатність створювати системне програмне забезпечення для кіберфізичних систем та веб-базованих систем Інтернету речей, мікроконтролерних і одноплатних комп'ютерних систем, а також програмованих користувачем вентильних матриць із використанням як низькорівневих, так і високорівневих мов програмування, реалізовувати сценарії автоматизованого управління пристроями, забезпечувати планування системних задач і підвищення надійності комп'ютерних систем (УК02).

програмних результатів навчання: знати новітні технології в галузі комп'ютерної інженерії (ПРН03); знати та розуміти вплив технічних рішень в суспільному, економічному, соціальному і екологічному контексті (ПРН04); мати знання основ економіки та управління проєктами (ПРН05); вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей (ПРН06); вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності (ПРН07); вміти системно мислити та застосовувати творчі здібності до формування нових ідей (ПРН08); вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання (ПРН10); вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії (ПРН11); вміти ефективно працювати як індивідуально, так і у складі команди (ПРН12); вміти виконувати експериментальні дослідження за професійною тематикою (ПРН15); вміти оцінювати отримані результати та аргументовано захищати прийняті рішення (ПРН16); спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською) (ПРН17); використовувати інформаційні технології та для ефективного спілкування на професійному та соціальному рівнях (ПРН18); здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення (ПРН19); усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення (ПРН20); розробляти системне програмне забезпечення та сценарії автоматизованого управління пристроями для кіберфізичних та веб-базованих систем Інтернету речей, мікроконтролерних і одноплатних комп'ютерних систем, зокрема Raspberry Pi, а також програмованих вентильних матриць (ПРН24).

Мета дисципліни. Формування у здобувачів вищої освіти компетентностей і практичних навичок щодо принципів та технологій об'єктно-орієнтованого програмування (ООП) з використанням мови програмування C++, а також розвиток умінь застосовувати ці знання для розв'язання інженерних та прикладних завдань у галузі комп'ютерної інженерії.

Предмет дисципліни. Методи, засоби та інструменти об'єктно-орієнтованого програмування мовою C++, що забезпечують побудову, моделювання, розробку та супровід ефективного, безпечного і масштабованого програмного забезпечення.

Завдання дисципліни. Формування практичних навичок застосування сучасних концепцій програмування та базовими принципами ООП (інкапсуляція, наслідування, поліморфізм); формування навичок проєктування програмних систем на основі класів, об'єктів та їх взаємодії; вивчення стандартних бібліотек C++ та інструментів для розробки програмних застосунків; набуття вмінь реалізовувати алгоритми та структури даних у об'єктно-орієнтованій парадигмі; розвиток компетентностей щодо аналізу, тестування та відлагодження програмного забезпечення; підготовка до використання ООП у суміжних дисциплінах та в подальшій професійній діяльності.

Результати навчання. Після вивчення дисципліни студент повинен: знати основні принципи об'єктно-орієнтованого програмування (інкапсуляція, наслідування, поліморфізм), синтаксис та можливості мови C++, уміти застосовувати сучасні методи програмування для розроблення алгоритмів, класів та об'єктів, використовувати стандартні бібліотеки та інструменти, мати навички аналізу, проєктування, реалізації та тестування програмних застосунків, у тому числі вбудованих і розподілених систем, вміти знаходити та критично оцінювати інформацію для розв'язання завдань, аргументувати вибір рішень, ефективно працювати індивідуально й у команді, оформлювати

результати у вигляді звітів та презентацій, усвідомлювати необхідність навчання протягом життя й бути готовим до застосування знань для професійної діяльності в галузі комп'ютерної інженерії.

4. Структура залікових кредитів дисципліни

Назва розділу (теми)	Кількість годин, відведених на:			
	лекції	практичні заняття	лабораторні заняття	СРС
Тема 1. Особливості мови програмування C++. Робота з файлами	2	4	4	18
Тема 2. Класи	2	8	8	36
Тема 3. Наслідування	2	4	4	18
Тема 4. Поліморфізм	2	4	4	18
Тема 5. Перевантаження операторів	2	2	2	18
Тема 6. Шаблони в C++	2	2	2	18
Тема 7. Бібліотека STL. Контейнери, ітератори, предикати в C++. Виключні ситуації	2	4	4	18
Тема 8. Обробка подій. Об'єктно-орієнтована програма, яка керована подіями	2	6	4	14
Разом за семестр:	16	34	32	158

5. Програма навчальної дисципліни

5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>Тема 1. Особливості мови програмування C++. Робота з файлами</i>	2
1	Особливості мови програмування C++. Робота з файлами. Локальні та глобальні змінні. Підпрограми та їх аргументи. Визначення даних. Модифікатор const. Модифікатор volatile. Перевантаження функцій. Потоки. Особливості роботи з файлами в C++ [1, 2, 3, 4, 6].	2
	<i>Тема 2. Класи</i>	2
2	Класи в C++. Поняття класу в C++. Поняття об'єкту та об'єктно-орієнтоване програмування. Визначення методів класу поза класом. Методи класу. Модифікатори доступу. Використання оператора глобального дозволу для елементів класу. Динамічний розподіл пам'яті. Ініціалізація виділеної пам'яті. Виділення пам'яті для масивів. Виділення пам'яті для об'єктів класів. Конструктори. Конструктори за замовчуванням. Конструктори з параметрами. Приватні конструктори. Конструктори копіювання. Деструктори. Загальнодоступні деструктори. Приватні деструктори. Статичні функції та елементи даних. Дружні функції та класи. Ключове слово this. Відношення між класами. Узагальнення. Асоціація. Агрегація. Композиція [1, 5, 7]	2
	<i>Тема 3. Наслідування.</i>	2
3	Наслідування. Поняття наслідування. Просте наслідування. Наслідування і захищені члени. Управління доступом до членів базового класу. Розв'язання конфлікту імен. Керування доступом до членів базового класу. Захищене наслідування. Конструктори, деструктори і наслідування. Передача параметрів конструктору базового класу. Надання доступу при наслідуванні. Множинне наслідування. Віртуальні класи при наслідуванні. [1, 6, 10, 11].	2
	<i>Тема 4. Поліморфізм</i>	2
4	Поліморфізм. Віртуальні функції та невіртуальні функції. Виклик віртуальної функції за допомогою посилання на об'єкт базового класу. Наслідування атрибуту virtual. поліморфізм. Абстрактні класи. Застосування віртуальних функцій. Чисто віртуальні функції. Віртуальні деструктори. Віртуальні конструктори [2,6,7,11].	2
	<i>Тема 5. Перевантаження операторів</i>	2
5	Перевантаження операторів Створення операторної функції-члена. Перевантаження скорочених операторів присвоєння. Обмеження на перевантажені оператори. Створення префіксної і постфіксної форм операторів інкрементації і декрементації. Перевантаження операторів за допомогою дружніх функцій. Застосування дружніх функцій для перевантаження операторів "++" і "--". Перевантаження операторів new і delete. Перевантаження деяких спеціальних операторів. [1,4,7,10].	2
	<i>Тема 6. Шаблони в C++</i>	2
6	Шаблони в C++ Шаблони функцій. Шаблони класів. Оголошення об'єктів, що базуються на шаблоні класу. [2,6,7,10].	2
	<i>Тема 7. Бібліотека STL. Контейнери, ітератори, предикати в C++. Виключні ситуації</i>	2
7	Стандартна бібліотека шаблонів STL. Виключні ситуації Складові STL. Класи-контейнери. Асоціативні контейнери. Алгоритми. Предикати. Застосування функторів. [1,9,12]. Основи обробки виняткових ситуацій. Перехоплення класів виняткових ситуацій. Обробка похідних виняткових ситуацій. Обмеження виключних ситуацій. Функції terminate() і unexpected(). [1, 8, 15]	2
	<i>Тема 8. Обробка подій. Об'єктно-орієнтована програма, яка керована подіями</i>	2
8	Обробка подій. Windows Form Applications Створення події. Обробка подій в C++. Компоненти, що утворюють інтерфейс між користувачем і додатком. Простір імен System. Основні компоненти. Властивості компонентів [6,7,13].	2
Разом за 2-й семестр		16

5.2 Зміст практичних занять

№ п/п	Тема практичного заняття	Кількість годин
	<i>Тема 1. Особливості мови програмування C++. Робота з файлами</i>	
1	Основні відмінності мови програмування C++ від Cі [1,4].	4
	<i>Тема 2. Класи</i>	
2	Класи, методи, оператори new, delete в C++[1-4].	4
3	Конструктори та деструктори, статичні функції та елементи даних, дружні функції та класи, конструктор копіювання в C++ [2,6,7,10].	4
	<i>Тема 3. Наслідування</i>	
4	Наслідування, множинне наслідування [2,6,7,10].	4
	<i>Тема 4. Поліморфізм</i>	
5	Поліморфізм, віртуальні функції, абстрактні класи та виняткові ситуації. [1-4,10].	4
	<i>Теми 5,6. Перевантаження операторів. Шаблони</i>	
6	Перевантаження операторів, шаблони в C++ [1,9,10].	4
	<i>Теми 7. Бібліотека STL. Контейнери, ітератори, предикати в C++. Виключні ситуації</i>	
7	Бібліотека STL. Контейнери, літератори, предикати в C++. [1,2,6,7].	4
	<i>Тема 8. Обробка подій. Об'єктно-орієнтована програма, яка керує подіями</i>	
8	Windows Form Applications [6,9,10]	6
	Разом:	34

5.3 Зміст лабораторних занять

№ п/п	Тема лабораторного заняття	Кількість годин
	<i>Тема 1. Особливості мови програмування C++. Робота з файлами</i>	
1	Основні відмінності мови програмування C++ від Cі [1,4].	4
	<i>Тема 2. Класи</i>	
2	Класи, методи, оператори new, delete в C++[1-4].	4
3	Конструктори та деструктори, статичні функції та елементи даних, дружні функції та класи, конструктор копіювання в C++ [2,6,7,10].	4
	<i>Тема 3. Наслідування</i>	
4	Наслідування, множинне наслідування [2,6,7,10].	4
	<i>Тема 4. Поліморфізм</i>	
5	Поліморфізм, віртуальні функції, абстрактні класи та виняткові ситуації. [1-4,10].	4
	<i>Тема 5. Перевантаження операторів. Шаблони</i>	
6	Перевантаження операторів, шаблони в C++ [1,9,10].	4
	<i>Теми 6,7. Бібліотека STL. Контейнери, ітератори, предикати в C++. Виключні ситуації</i>	
7	Бібліотека STL. Контейнери, літератори, предикати в C++. [1,2,6,7].	4
	<i>Тема 8. Обробка подій. Об'єктно-орієнтована програма, яка керує подіями</i>	
8	Windows Form Applications [6,9,10]	4
	Разом:	32

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

Самостійна робота студентів усіх форм здобуття освіти полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації, підготовці до практичних занять, контрольних робіт, підготовці до виконання і захисту лабораторних робіт, тестування. Крім 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
1-2	Опрацювання лекційного матеріалу. Підготовка до ЛР1 та ПР1. Підготовка до захисту ЛР1.	18
3-4	Опрацювання лекційного матеріалу. Підготовка до ЛР2 та ПР2. Підготовка до захисту ЛР2.	18
5-6	Опрацювання лекційного матеріалу. Підготовка до ЛР3 та ПР3. Підготовка до захисту ЛР3.	18
7-8	Опрацювання лекційного матеріалу. Підготовка до ЛР4 та ПР4. Підготовка до захисту ЛР4.	18
9-10	Опрацювання лекційного матеріалу. Підготовка до ЛР5 та ПР5. Підготовка до захисту ЛР5.	18
11-12	Опрацювання лекційного матеріалу. Підготовка до ЛР6 та ПР6. Підготовка до захисту ЛР6.	18
13-14	Опрацювання лекційного матеріалу. Підготовка до ЛР7 та ПР7. Підготовка до захисту ЛР7.	18
15-16	Опрацювання лекційного матеріалу. Підготовка до ЛР8 та ПР8. Підготовка до захисту ЛР8.	18
17	Опрацювання лекційного матеріалу. Підготовка до тестування.	14
Разом семестр:		158

Примітки: ЛР– лабораторна робота, ПР – практична робота.

6. Технології та методи навчання

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій та методів навчання, зокрема: лекції (з використанням методів візуалізації, проблемного й інтерактивного навчання, мотиваційних прийомів, інформаційно-комунікаційних технологій); практичні заняття (з використанням інструктування, демонстрування, розв'язування типових і прикладних задач, аналізу кейсів, ситуаційних завдань, елементів дискусії тощо); лабораторні заняття (з використанням методів комп'ютерного моделювання, методів проектної діяльності); самостійна робота (опрацювання теоретичного матеріалу, підготовка до виконання практичних робіт, поточного та підсумкового контролю), з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

7. Методи контролю

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

- тестовий контроль засвоєння теоретичного та практичного матеріалу з теми;
- оцінювання контрольних робіт (практичних завдань за темами).
- оцінювання результатів захисту лабораторних робіт;
- оцінювання результатів роботи на практичних заняттях
- тестовий контроль теоретичного матеріалу з розділу;
- оцінювання контрольних робіт.

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; відвідування занять з дисципліни є обов'язковим. За об'єктивних причин (підтверджених документально) теоретичне навчання за погодженням із лектором може відбуватись в он-лайн режимі. Успішне опанування дисципліни і формування фахових компетентностей і програмних результатів навчання передбачає необхідність підготовки до практичних занять (вивчення теоретичного матеріалу з теми), активно працювати на занятті, коректно виконувати практичні завдання з об'єктно-орієнтованого програмування, брати участь у дискусіях щодо прийнятих рішень при виконанні здобувачами задач; необхідність підготовки до лабораторних занять (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)), якісно підготувати звіт, захистити результати виконаної роботи.

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 практичне або 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 практичних занять, тестування та контрольної роботи.

Здобувач вищої освіти, виконуючи самостійну роботу або індивідуальну роботу з дисципліни, має дотримуватися політики доброчесності (заборонені списування, плагіат (в т.ч. із 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 і зарахування результатів навчання, набутих шляхом неформальної освіти, що розміщені на доступних платформах (<https://www.coursera.org/learn/c-plus-plus-a?specialization=coding-for-everyone>), які сприяють формуванню компетентностей і 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 критерія 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 і 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 і 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, допускається помилок у визначенні понять, перекошує 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	
3 семестр														
Лабораторні роботи №:								Практична роботи №:			Тестовий контроль ТК		Іспит	Разом балів
1	2	3	4	5	6	7	8	1-2	3-5	6-8	1			
Кількість балів за вид навчальної роботи (мінімум-максимум)														
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5		24-40	
24-40								9-15			3-5		24-40	60-100*

Примітки: *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 виконаного завдання, відповіді та знання з об'єктно-орієнтованого програмування; дотримання вимог при оформленні звіту з лабораторної роботи.

Результат виконання і захисту здобувачем вищої освіти кожної лабораторної роботи оцінюється відповідно до таблиці Критеріїв оцінювання навчальних досягнень здобувача вищої освіти.

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, методику її виконання, виправити грубі помилки та повторно вийти на її захист у призначений для цього 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 3 до 5 балів.

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

Кількість правильних відповідей	1-13	14-17	19-20	21-25
Відсоток правильних відповідей	0-59	60-72	73-82	83-100
Кількість балів	-	3	4	5

На тестування відводиться 40 хвилин. Студент проходить тестування в он-лайн 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, 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	3	4	5
Теоретичне питання № 2	3	4	5
Практичне завдання	18	24	30
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 вище (**Таблиця – 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 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)	
		Залік	Іспит/диференційований залік
A	90-100	Зараховано	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	Незараховано	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

- Загальні відомості про програми на C++. Структура проектів в C++.
- Елементи мови. Службові слова. Імена. Коментарі.
- Розміщення даних у пам'яті. Тип даних. Адресний вираз.
- Функції. Область існування імені. Область видимості та простору імен.
- Типи. Базові стандартні типи. Константи. Змінні.
- Операції C++. Арифметичні операції. Операції присвоєння. Операції відношення та еквівалентності. Логічні операції.
- Операції адресації та непрямої адресації.
- Пріоритет і асоціативність операцій. Перегрузка операцій.
- Оператори передачі управління. Умовний оператор множинного вибору.
- Оператори циклів: for, do...while, while.
- Переривання циклу: оператори break, continue, return, функція Abort.
- Умовна компіляція. Операції препроцесора.
- Масиви у C++. Операції з масивами.
- Правила роботи з масивами.
- Структури та об'єднання. Структури в стилі C++.
- Об'єднання. Об'єднання, що не мають імені. Доступ до членів-даних структури, об'єднання.
- Екземпляри структури, об'єднання. Масиви структур.
- Вказівники, зсилки та функції в C++.
- Локальний та динамічний розподіл пам'яті. Функції malloc, calloc, realloc, free.
- Оператори new и delete. Динамічне розміщення об'єктів та вказівників.
- Правила роботи з вказівниками. Розіменування вказівників.
- Оператори new[], оператори delete[].
- Функції в C++. Передача аргументів та повернення результату.
- Застосування при передачі параметрів специфікації const. Параметри зі значеннями по умовчанняю.
- Перегрузка функцій. Приведення типів.
- Функції-члени класу. Передача у функції змінного числа параметрів.
- Вбудовані функції inline.
- Шаблони функцій.
- Область видимості функцій. Правила, що визначають область видимості.

30. Функція main(). Аргументи функції main. Аргументи командного рядка.
31. Основні поняття ООП. Інкапсуляція, поліморфізм, наслідування і засоби їх реалізації.
32. Об'єкт, властивості, методи, події.
33. Класи. Визначення класу. Ідентифікатори класу. Тіло класу.
34. Приховування інформації. Ієрархії класів та наслідування.
35. Інкапсуляція. Керування доступом до класу. Приватні, загальнодоступні, захищені члени класу.
36. Класи пам'яті для об'єктів класів. Область видимості класу. Порожні класи.
37. Вкладені класи. Правила доступу для вкладених класів.
38. Екземпляри класу. Використання членів даних. Статичні члени-данні.
39. Об'єкти класу як члени даних. Вказівники як члени даних.
40. Використання функцій-членів. Прості, статичні функції члени.
41. Конструктори. Деструктори.
42. Дружні функції. Властивості дружніх функцій.
43. Шаблони класів. Вкладені шаблонні класи.
44. Наслідування. Обмеження наслідування. в C++. Просте наслідування.
45. Специфікатори доступу базових класів. Класи для наслідування. Аргументи передані в базовий клас.
46. Порядок виклику конструкторів. Порядок виклику деструкторів.
47. Перетворення типів у похідних класах. Дозвіл області видимості.
48. Множинне наслідування. Оголошення класу з декількома базовими класами. Виклик конструкторів базових класів.
49. Використання віртуальних базових класів. Використання віртуальних і невіртуальних базових класів разом.
50. Використання перетворення типів. Правила виклику функцій базових класів.
51. Використання дозволу видимості при множинному наслідуванні.
52. Поліморфізм. Раннє та пізнє зв'язування. Віртуальні функції. Перевизначення функції.
53. Абстрактні класи. Обмеження віртуальних функцій. Віртуальні оператори.
54. Поліморфізм при простому наслідуванні. Поліморфізм при множинному наслідуванні.
55. Виклик поліморфних функцій базового класу. Віртуальні функції та ієрархії класифікацій. Виклик віртуальних функцій у конструкторах.
56. Потік C++. Потоки як узагальнені фільтри. Стандартний потоковий ввід-вивід класів користувача.
57. Потоки і файловий ввід-вивід. Використання текстових файлів для введення. Вивід текстових файлів.
58. Обробка подій. Об'єктно-орієнтована програма, яка керована подіями. Методи опрацювання подій.
59. Шаблони функцій і класів. Основні властивості шаблонів класів. Компонентні функції параметризованих класів.
60. Стандартна бібліотека шаблонів. Склад STL. Класи-контейнери. Асоціативні контейнери. Алгоритми.

11. Навчально-методичне забезпечення

Освітній процес з дисципліни «Об'єктно-орієнтоване програмування» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлені і видані такі роботи:

1. Об'єктно-орієнтоване програмування : методичні вказівки до виконання лабораторних робіт /С. М. Лисенко. – Хмельницький : ХНУ, 2020. – 200 с.

12. Матеріально-технічне та програмне забезпечення дисципліни (за потреби)

Інформаційна та комп'ютерна підтримка: ПК, планшет, смартфон або інший мобільний пристрій, проектор. Програмне забезпечення: програми Microsoft Office або аналогічні, доступ до мережі Інтернет, робота з презентаціями.

Вивчення навчальної дисципліни не потребує використання спеціального програмного прикладного забезпечення, крім загальноновживаних програм і операційних систем.

13. Рекомендована література:

Основна

1. Єфіменко С.В. Методичний посібник з курсу «об'єктно-орієнтоване програмування. Мови C/C++. Київ. 2021. 124 с.
2. Данілова В.А. Об'єктно-орієнтоване програмування. Практикум. Київ КПІ ім. Ігоря Сікорського 2021. с.121.
3. Порев В.М. Об'єктно-орієнтоване програмування: конспект лекцій : навч. посіб. для студ. освітньої програми «Інженерія програмного забезпечення комп'ютерних систем» спеціальності 121 «Інженерія програмного забезпечення»; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2022. 271 с.
4. Бобков, В. Б., Грудзинський Ю. Є., Крилов К. В. Об'єктно-орієнтоване програмування. Навчальний посібник : навчальний посібник ; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2023. 77 с.
5. Омельчук Л.Л., Белова А.С. Об'єктно-орієнтоване програмування. Лабораторний практикум: навчальний посібник. Київ : 2022. 271 с.
6. Зеленський О.С., Лисенко В.С. Об'єктно-орієнтоване програмування на C++. Навчальний посібник. Кривий Ріг: Державний університет економіки і технологій, 2023. 215 с.
7. Гришанович Т. О., Глинчук Л. Я. Основи об'єктно-орієнтованого програмування : навч. посібник. Луцьк : ВНУ імені Лесі Українки, 2022. –120 с.

8. Об'єктно-орієнтоване програмування мовою C# : навч. посіб. для здобувачів першого (бакалавр.) рівня вищ. пед. освіти, пед. кадрів закл. заг. серед. освіти / Н. В. Олефіренко, А. Р. Курганський, Л. П. Остапенко, Н. О. Пономарьова ; Харків. нац. пед. ун-т ім. Г. С. Сковороди. – Харків : [б. в.], 2024. – 253 с. : іл., табл.
9. Stroustrup, B. (2024). *Programming: Principles and Practice Using C++*. Addison-Wesley Professional.
10. Mak, R. (2024). *Object-oriented Software Design in C++*. Simon and Schuster.
11. Bhatt, A., Kuksal, R., & Shah, P. (2025). *Programming Principles and Practice Using C++*. *ASP Books*, 01-93.
12. Rangiseti, A. K. (2024). Start Learning OOP Using C++. In *Hands-On Object-Oriented Programming: Mastering OOP Features for Real-World Software Systems Development* (pp. 43-111). Berkeley, CA: Apress.
13. Singhal, K., & Bhatt, A. (2025). *Advanced Object Oriented Programming: Solving Real-World Problems*. *ASP Books*, 01-260.
14. Campbell, O. O., Bake, P., Mohammed, A. A., & Dagah, A. I. (2024). *A Guided Introduction to C++ Programming*.

14. Інформаційні ресурси

1. Модульне середовище для навчання. URL : <https://msn.khmnu.edu.ua/course/view.php?id=67>
2. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
3. Інституційний репозитарій ХНУ. URL : <https://clar.khmnu.edu.ua/home>

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	Перший
Кількість призначених кредитів ЄКТС	8,0
Форми здобуття освіти, для яких викладається дисципліна	Очна (денна)

Результати навчання. Після вивчення дисципліни студент повинен: знати основні принципи об'єктно-орієнтованого програмування (інкапсуляція, наслідування, поліморфізм), синтаксис та можливості мови C++, уміти застосовувати сучасні методи програмування для розроблення алгоритмів, класів та об'єктів, використовувати стандартні бібліотеки та інструменти, мати навички аналізу, проєктування, реалізації та тестування програмних застосунків, у тому числі вбудованих і розподілених систем, вміти знаходити та критично оцінювати інформацію для розв'язання завдань, аргументувати вибір рішень, ефективно працювати індивідуально й у команді, оформлювати результати у вигляді звітів та презентацій, усвідомлювати необхідність навчання протягом життя й бути готовим до застосування знань для професійної діяльності в галузі комп'ютерної інженерії.

Зміст навчальної дисципліни. Особливості мови програмування C++. Робота з файлами. Класи. Наслідування. Поліморфізм. Перевантаження операторів. Шаблони в C++. Бібліотека STL. Контейнери, ітератори, предикати в C++. Виключні ситуації. Обробка подій. Об'єктно-орієнтована програма, яка керована подіями.

Пререквізити – Програмування (ОПП.01)

Постреквізити – Бази даних (ОФП.04), Кваліфікаційна робота (ОФП.23).

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); практичні заняття (з використанням інструктування, демонстрування, розв'язування типових і прикладних задач, аналізу кейсів, ситуаційних завдань, елементів дискусії тощо); самостійна робота (опрацювання теоретичного матеріалу, підготовка до виконання практичних робіт, поточного та підсумкового контролю, виконання індивідуальних та домашніх завдань), з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

Форми оцінювання результатів навчання: оцінювання практичних, лабораторних, контрольної роботи; тестування.

Вид семестрового контролю: іспит – 1 семестр

Навчальні ресурси:

1. Данілова В.А. Об'єктно-орієнтоване програмування. Практикум. Київ КПІ ім. Ігоря Сікорського 2021. с.121.
2. Порєв В.М. Об'єктно-орієнтоване програмування: конспект лекцій : навч. посіб. для студ. освітньої програми «Інженерія програмного забезпечення комп'ютерних систем» спеціальності 121 «Інженерія програмного забезпечення»; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2022. 271 с.
3. Об'єктно-орієнтоване програмування мовою C# : навч. посіб. для здобувачів першого (бакалавр.) рівня вищ. пед. освіти, пед. кадрів закл. заг. серед. освіти / Н. В. Олефіренко, А. Р. Курганський, Л. П. Остапенко, Н. О. Пономарьова ; Харків. нац. пед. ун-т ім. Г. С. Сковороди. – Харків : [б. в.], 2024. – 253 с. : іл., табл.
4. Stroustrup, B. (2024). Programming: Principles and Practice Using C++. Addison-Wesley Professional.
5. Mak, R. (2024). Object-oriented Software Design in C++. Simon and Schuster.
6. Bhatt, A., Kuksal, R., & Shah, P. (2025). Programming Principles and Practice Using C++. ASP Books, 01-93.
7. Модульне середовище для навчання. URL : <https://msn.khmnmu.edu.ua/course/view.php?id=67>
8. Електронна бібліотека ХНУ. Доступ до ресурсу: <http://library.khmnmu.edu.ua/>

Викладач: д-р. техн. наук, професор Лисенко С.М.