

Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Спеціальність **121 Інженерія програмного забезпечення**

Освітня програма «**Інженерія програмного забезпечення**»

Рівень вищої освіти **перший (бакалаврський)**

Факультет математики та інформатики

2024-2025 навчальний рік

Робоча програма навчальної дисципліни «Об'єктно-орієнтоване програмування» для здобувачів першого (бакалаврського) рівня вищої освіти зі спеціальності 121 Інженерія програмного забезпечення за освітньою програмою «Інженерія програмного забезпечення»

Мова навчання: українська

Розробники: Бичков О.С., професор кафедри інформаційних технологій та моделювання,
докт. тех. наук, професор;
Кирик Т.А., старший викладач кафедри інформаційних технологій та моделювання.

Робоча програма затверджена на засіданні кафедри інформаційних технологій та моделювання.

Протокол від 3 жовтня 2024 року № 9-А.

Завідувач кафедри



Мороз І. П.

Робочу програму схвалено навчально-методичною комісією факультету математики та інформатики.

Протокол від 4 жовтня 2024 року № 7-Б.

Голова навчально-методичної комісії



Білецький В. В.

© Бичков О.С., 2024 р.
© Кирик Т.А., 2024 р.
© РДГУ, 2024 р.

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів:	Галузь знань: 12 Інформаційні технології Спеціальність: 121 Інженерія програмного забезпечення Освітня програма: «Інженерія програмного забезпечення» Рівень вищої освіти: перший (бакалаврський)	Обов'язкова	Обов'язкова
Модулів:		Рік підготовки:	
Змістових модулів:		2-й	2-й
Індивідуальне науково-дослідне завдання:		Семестр:	
об'єктно-орієнтована програмна модель		3-й	3-й
Загальна кількість годин:		Лекції:	
Тижневих годин:		26 год.	8 год.
аудиторних:		Практичні:	
самостійної роботи студента:		-	-
		Лабораторні:	
		24 год.	8 год.
		Самостійна робота:	
		100	134
		Індивідуальні завдання:	
		-	-
	Вид контролю:		
	екзамен	екзамен	

Передумови для вивчення дисципліни: «Програмування», «Алгоритми і структури даних».

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Навчальна дисципліна «Об'єктно-орієнтоване програмування» відноситься до обов'язкових компонентів професійної підготовки здобувачів вищої освіти першого (бакалаврського) рівня за спеціальністю 121 Інженерія програмного забезпечення. Робоча програма навчальної дисципліни складена у відповідності до освітньо-професійної програми «Інженерія програмного забезпечення» підготовки бакалаврів за названою спеціальністю.

Метою викладання дисципліни «Об'єктно-орієнтоване програмування» є формування у студентів розуміння об'єктно-орієнтованої парадигми аналізу, проектування і програмування в предметних областях інженерії програмного забезпечення та вміння ефективно застосовувати їх на практиці при проектуванні, реалізації та тестуванні програмного забезпечення.

Основними **завданнями** дисципліни «Об'єктно-орієнтоване програмування» є оволодіння здобувачами вищої освіти методологією об'єктно-орієнтованого програмування, формування практичних умінь і навичок проектування та програмної реалізації класів і об'єктів на основі інкапсуляції, успадкування, поліморфізму, абстракції та інших визначальних принципів ООП для створення гнучких, легко підтримуваних і розширюваних програм.

Згідно з освітньо-професійною програмою навчальна дисципліна «Об'єктно-орієнтоване програмування» має забезпечити формування у здобувачів вищої освіти відповідних **компетентностей**.

Загальні компетентності

- K01. Здатність до абстрактного мислення, аналізу та синтезу.
- K02. Здатність застосовувати знання у практичних ситуаціях.

K05. Здатність вчитися і оволодівати сучасними знаннями.

Фахові компетентності

K20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

K22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

K25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

K26. Здатність до алгоритмічного та логічного мислення.

K28. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.

3. ОЧІКУВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

У результаті освоєння повного курсу навчальної дисципліни «Об'єктно-орієнтоване програмування» у здобувачів вищої освіти формуються глибокі, міцні і системні знання, які передбачають вільне володіння понятійним апаратом, розуміння основних задач предмету, його мети та завдання, а також здатність до практичного застосування цих знань при реалізації прикладних застосувань. Згідно з освітньо-професійною програмою мають бути досягнуті наступні **програмні результати навчання**:

ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПР05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПР15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПР26. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі інженерії програмного забезпечення.

Здобувачі вищої освіти повинні

знати:

- визначальні принципи ООП: інкапсуляція, успадкування, поліморфізм та абстракція;
- особливості реалізації принципів ООП у мові програмування C++ при конструюванні класів та створенні об'єктів;
- відношення між класами: асоціація, агрегація, композиція;
- принципи переважання операторів;
- особливості конструювання шаблонів класів;
- компоненти стандартної бібліотеки шаблонів STL;

вміти:

- здійснювати розробку та реалізацію об'єктно-орієнтованих моделей систем із використанням мови C++;
- використовувати успадкування, поліморфізм, інтерфейси та абстрактні класи;

- проектувати взаємодію між класами з урахуванням різних типів зв'язків: асоціація, агрегація, композиція;
- використовувати механізми перевантаження операторів для спрощення роботи з класами;
- застосовувати шаблони класів для узагальнення та повторного використання коду;
- використовувати компоненти STL для роботи з контейнерами, алгоритмами та ітераторами;
- виконувати тестування та налагодження програм;
- проводити аналіз, оптимізацію та вдосконалення створених програм.

4. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Змістовий модуль 1. «Основи ООП в C++. Класи та об'єкти»

Тема 1. Основні принципи об'єктно-орієнтованого програмування. Об'єктно-орієнтоване програмування у порівнянні з процедурним. Поняття класу, об'єкта. Стан і поведінка об'єкта. Інкапсуляція.

Тема 2. Класи та об'єкти. Протокол опису класу. Створення та ініціалізація об'єктів. Доступ до полів та методів. Використання специфікаторів доступу класу. Перевантаження функцій–елементів класу. Конструктори. Перевантажені конструктори. Список ініціалізації конструктора. Делегування конструкторів. Деструктори.

Тема 3. Поверхнєве та глибинне копіювання. Конструктори копії, перетворення. Вказівник *this*. Константні функції елементи. Константні об'єкти. Вказівники на об'єкти. Динамічні об'єкти та масиви об'єктів.

Тема 4. Статичні елементи класу. Статичні змінні та функції класу. Застосування.

Тема 5. Друзі класу. Дружні функції класу та їх призначення. Дружні класи та їх використання.

Тема 6. Перевантаження операцій. Перевантаження операцій унарних, бінарних. Перевантажені операції як дружні функції та як функції–елементи класу. Перевантаження операцій зчитування з потоку вводу та вставки у потік виводу.

Змістовий модуль 2. «Відношення між класами»

Тема 7. Відношення між класами. Відношення асоціації. Агрегація об'єктів. Композиція об'єктів.

Тема 8. Успадкування. Механізм успадкування. Керування доступом при успадкуванні. Множинне успадкування. Конфлікти при успадкуванні. Проблема ромба. Віртуальний базовий клас.

Тема 9. Поліморфізм. Статичний та динамічний поліморфізм. Раннє та пізнє зв'язування. Віртуальні функції. Віртуальний деструктор. Чисті віртуальні функції та абстрактні класи. Чистий віртуальний деструктор. Інтерфейси.

Тема 10. Потоківі класи. Ієрархія потікових класів C++. Файлові потоки. Рядкові потоки.

Змістовий модуль 3. «Розширені можливості ООП в C++»

Тема 11. Механізм обробки винятків. Створення та використання власних типів винятків. Робота із стандартними винятками.

Тема 12. Шаблони класів. Стандартні шаблонні типи *pair<>*, *tuple<>*, *initializer_list<>*.

Тема 13. Бібліотека STL. Поняття контейнера, алгоритму, ітератора. Використання контейнерів: вектор, список, дек, черга, стек, множина, карта. Алгоритми роботи з контейнерами.

5. СТРУКТУРА НАВЧАЛЬНОЇ ПРОГРАМИ

Назви змістових модулів і тем	Кількість годин											
	денна форма навчання						заочна форма навчання					
	усього	у тому числі					усього	у тому числі				
		лек	пр	лаб	інд	с.р.		лек	пр	лаб	інд	с.р.
Модуль 1												
Змістовий модуль 1. «Основи ООП в C++. Класи та об'єкти»												
Тема 1. Основні принципи об'єктно-орієнтованого програмування.	12	2		2		8	12	1				11
Тема 2. Класи та об'єкти.	12	2		2		8	12			1		11
Тема 3. Поверхнєве та глибинне копіювання. Масиви об'єктів	12	2		2		8	12	1				11
Тема 4. Статичні елементи класу	9	1		2		6	9			1		8
Тема 5. Друзі класу.	9	1		2		6	9	1				8
Тема 6. Перевантаження операцій.	12	2		2		8	12			1		11
Разом за змістовим модулем 1	66	10	-	12	-	44	66	3	-	3	-	60
Змістовий модуль 2. «Відношення між класами»												
Тема 7. Відношення між класами.	12	2		2		8	12	1				11
Тема 8. Успадкування.	15	3		2		10	15	1		1		13
Тема 9. Поліморфізм.	15	3		2		10	15	1		1		13
Тема 10. Потоківі класи у C++.	6	2				4	6			1		5
Разом за змістовим модулем 2	48	10	-	6	-	32	48	3		3		42
Змістовий модуль 3. «Розширені можливості ООП в C++»												
Тема 11. Механізм обробки винятків.	12	2		2		8	12	1				11
Тема 12. Шаблони класів.	9	1		2		6	9			1		8
Тема 13. Бібліотека STL.	15	3		2		10	15	1		1		13
Разом за змістовим модулем 3	36	6		6		24	36	2		2		32
Модуль 2												
ІНДЗ												
Усього годин	150	26	-	24	-	100	150	8		8		134

6. ТЕМИ СЕМІНАРСЬКИХ ЗАНЯТЬ

Не передбачено навчальним планом

7. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

Не передбачено навчальним планом

8. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з.п.	Назва теми	Кількість год.
1.	Класи та об'єкти. Дані–елементи та функції–елементи класу. Доступ до елементів класу. Функції доступу.	2
2.	Конструктори та деструктори. Конструктори по замовчуванню.	2

	Перевантажені конструктори. Вказівник <i>this</i> .	
3.	Конструктори копії. Конструктори перетворення. Перевантаження функцій-елементів класу. Постійні методи класу. Масиви об'єктів. Вказівники на об'єкти. Динамічні об'єкти та масиви об'єктів.	2
4.	Статичні змінні та функції класу.	2
5.	Дружні класу функції. Дружні класи.	2
6.	Перевантаження операцій (арифметичних, відношення, логічних, інкременту, декременту, присвоєння, приведення типу, індексування, вставки у потік виводу та зчитування з потоку вводу).	2
7.	Асоціація. Агрегація та композиція об'єктів.	2
8.	Успадкування. Одиночне успадкування. Керування доступом до елементів класу при спадкуванні. Множинне успадкування. Виклик конструкторів та деструкторів. Проблема ромба.	2
9.	Статичний та динамічний поліморфізм. Раннє та пізнє зв'язування. Віртуальні функції-елементи. Абстрактний базовий клас.	2
10.	Механізм обробки винятків	2
11.	Шаблони класів.	2
12.	STL	2
Разом		24

САМОСТІЙНА РОБОТА

№ з.п.	Назва теми	Кількість годин	
		д.ф.н.	з.ф.н.
1.	Основні принципи ООП.	5	8
2.	Порядок виклику конструкторів, деструкторів для локальних, глобальних та динамічних об'єктів. Делегування конструкторів. Список ініціалізації конструктора.	5	8
3.	Поверхнєве та глибинне копіювання. Масиви об'єктів. Конструктор копії.	5	8
4.	Статичні дані та методи класу. Приклади застосувань	5	5
5.	Константи класу. Вказівники на функції-елементи класу.	4	3
6.	Масиви об'єктів. Ініціалізація. Виклики конструкторів та деструкторів.	4	5
7.	Дружні функції, методи, класи.	4	6
8.	Перевантаження операцій, операцій доступу до елемента структури по вказівнику (->), розіменування вказівника (*).	10	15
9.	Асоціація. Агрегація об'єктів. Композиція об'єктів. Приклади застосування.	6	9
10.	Ієрархії класів та об'єктів в ООП. Наслідування як механізм породження ієрархій класів. Проблема ромба. Віртуальні базові класи.	8	12
11.	Статичний та динамічний поліморфізм. Таблиця віртуальних функцій. Віртуальні деструктори. Абстрактні базові класи. Інтерфейси	10	15
12.	Файлові потоки. Рядкові потоки.	8	8
13.	Обробка винятків. Стандартні класи винятків. Створення користувацьких класів винятків.	8	10
13.	Шаблони класів. Стандартні шаблонні типи <i>pair<></i> , <i>tuple<></i> ,	6	7

№ з.п.	Назва теми	Кількість годин	
		д.ф.н.	з.ф.н.
	<i>initializer_list</i> <>.		
14.	Бібліотека STL. Контейнери, ітератори, алгоритми. Види ітераторів.	12	19
Разом		100	138

10. ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Варіанти індивідуального завдання передбачають розробку програмного продукту або окремого модуля, що демонструє використання принципів об'єктно-орієнтованого програмування, реалізацію успадкування, поліморфізму, асоціації, агрегації чи композиції, перевантаження операцій, а також застосування бібліотеки STL.

11. МЕТОДИ НАВЧАННЯ.

- МН1 – словесний метод (лекція, дискусія, обговорення досліджуваного явища чи процесу, аналіз проблемних ситуацій);
МН2 – практичний метод (лабораторні заняття);
МН3 – наочний метод (ілюстрації, демонстрації);
МН4 – робота з навчально-методичною літературою (конспектування, тезування, складання реферату);
МН5 – інтерактивний метод (із застосуванням аудіо, відео, новітніх інформаційних технологій та комп'ютерних засобів навчання);
МН6 – самостійна робота (самостійний і/або груповий аналіз, проектування та реалізація задач);
МН7 – індивідуальна науково-дослідна робота здобувачів вищої освіти (розв'язування прикладних задач із реальної предметної області).

12. МЕТОДИ ОЦІНЮВАННЯ

- МО1 – екзамен;
МО2 – усне або письмове опитування;
МО4 – тестування;
МО7 – презентація та обговорення результатів виконаних індивідуальних науково-дослідних завдань та досліджень;
МО8 – перевірка контрольних робіт;
МО9 – захист лабораторних робіт.

13. ЗАСОБИ ДІАГНОСТИКИ РЕЗУЛЬТАТІВ НАВЧАННЯ

- презентації результатів виконаних завдань лабораторних робіт;
- тести поточного та підсумкового контролю знань;
- завдання модульної контрольної роботи;
- тематика і завдання індивідуальні науково-дослідної роботи;
- екзамен.

Види та методи навчання і оцінювання

Код ЗК, СК	Назва компетентності	Код ПРН	Назва програмного результату навчання	Методи навчання	Методи оцінювання
К01	Здатність до абстрактного мислення, аналізу	ПР05	Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-	МН1, МН2, МН3, МН4, МН5, МН6,	МО1, МО2, МО4, МО7, МО8, МО9

	та синтезу.		орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.	МН7	
		ПР07	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
		ПР13	Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
K02	Здатність застосовувати знання у практичних ситуаціях.	ПР05	Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
		ПР07	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
		ПР13	Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
K05	Здатність вчитися і оволодівати сучасними знаннями.	ПР01	Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
K20	Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.	ПР01	Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
		ПР05	Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
		ПР07	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
K22	Здатність накопичувати, обробляти та систематизувати професійні знання	ПР01	Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9

	щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.		науки і техніки.		
		ПР15	Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
		ПР26	Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі інженерії програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
K25	Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.	ПР15	Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
		ПР26	Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі інженерії програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
K26	Здатність до алгоритмічного та логічного мислення.	ПР05	Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
		ПР13	Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
		ПР26	Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі інженерії програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
K28	Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного	ПР13	Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9
		ПР26	Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі інженерії програмного забезпечення.	МН1, МН2, МН3, МН4, МН5, МН6, МН7	МО1, МО2, МО4, МО7, МО8, МО9

	програмного забезпечення.				
--	---------------------------	--	--	--	--

14. КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Результат освітньої діяльності здобувача вищої освіти оцінюється згідно Положення про оцінювання знань і умінь здобувачів вищої освіти РДГУ за такими критеріями оцінювання та рівнями компетентності:

Суми балів за 100-бальною шкалою	Оцінка в ЄКТС	Значення оцінки ЄКТС	Критерії оцінювання	Рівень компетентності	Оцінка за національною шкалою
90-100	A	відмінно	здобувач вищої освіти виявляє особливі творчі здібності, вміє самостійно здобувати знання, без допомоги викладача знаходить і опрацьовує необхідну інформацію, вміє використовувати набуті знання і вміння для прийняття рішень у нестандартних ситуаціях, переконливо аргументує відповіді, самостійно розкриває власні здібності	високий (творчий)	відмінно
82-89	B	добре	здобувач вищої освіти вільно володіє теоретичним матеріалом, застосовує його на практиці, вільно розв'язує вправи і задачі у стандартних ситуаціях, самостійно виправляє допущені помилки, кількість яких незначна	достатній (конструктивно-варіативний)	добре
74-81	C	добре	здобувач вищої освіти вміє зіставляти, узагальнювати, систематизувати інформацію під керівництвом викладача, загалом самостійно застосовувати її на практиці; контролювати власну діяльність; виправляти помилки, з-поміж яких є суттєві, добирати аргументи для підтвердження думок		
64-73	D	задовільно	здобувач вищої освіти відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, за допомогою викладача може аналізувати навчальний матеріал, виправляти помилки, з-поміж яких є значна кількість суттєвих	середній (репродуктивний)	задовільно
60-63	E	задовільно	здобувач вищої освіти володіє навчальним матеріалом на		

			рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні		
35-59	FX	незадовільно з можливістю повторного складання семестрового контролю	здобувач вищої освіти володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу	низький (рецептивно-продуктивний)	незадовільно
1-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	здобувач вищої освіти володіє матеріалом на рівні елементарного розпізнавання і відтворення окремих фактів, елементів, об'єктів	низький (рецептивно-продуктивний)	незадовільно

Підсумкова (загальна) оцінка з навчальної дисципліни є сумою рейтингових оцінок (балів), одержаних за окремі оцінювальні форми навчальної діяльності: поточне і підсумкове оцінювання рівня засвоєння теоретичного та практичного матеріалу під час аудиторних занять і самостійної роботи; оцінка (бали) за виконання лабораторних завдань; оцінка (бали) за індивідуальну науково-дослідну роботу; оцінка (бали) за участь у наукових конференціях, олімпіадах, підготовку наукових публікацій, рефератів тощо.

15. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ ВИЩОЇ ОСВІТИ

В університеті діє накопичувальна кредитно-трансферна система оцінювання програмних результатів навчання студентів, що реалізується в ході виконання і захисту лабораторних робіт, виконання ІНДЗ та модульного контролю, для яких визначено мінімальну кількість балів, яку слід набрати для формування рейтингового балу студента та виставлення його у залікову книжку і відомість успішності студентів з відповідними оцінками за національною та європейською кредитно-трансферною системами (ЄКТС).

Розподіл балів за видами освітньої діяльності

П'ятибальна система оцінок	: 3, 4, 5
Захист лабораторних робіт	: 1, 2, 3
Тестування	: 2, 4, 6
Контрольна робота	: 2, 4, 6
ІНДЗ	: 6, 8, 10
Екзамен	: до 40

Види навчальної діяльності	Модуль 1												Модуль 2	Екзамен	Сума	
	Змістовий модуль 1						Змістовий модуль 2				Змістовий модуль 3					
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12				T13
Захист лаб. робіт	3	3	3	3	3	3	3	3	3	2	3	3	3	ІНДЗ	Екзамен	Сума
Контроль за сам. роб.	Тест / контрольна робота 6						Тест / контрольна робота 6									
Разом	24						26						10			

T1, T2, ..., T13 – теми змістових модулів.

16. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

В якості навчально-методичного забезпечення самостійної роботи студентів використовується базова та додаткова література з дисципліни, Інтернет-ресурси, матеріал лекцій, методичні рекомендації для виконання лабораторних робіт.

17. ПИТАННЯ ДЛЯ ПІДГОТОВКИ ДО ПІДСУМКОВОГО КОНТРОЛЮ

1. Об'єктно-орієнтоване програмування у порівнянні з процедурним. Основні принципи об'єктно-орієнтованого програмування.
2. Основні принципи ООП.
3. Поняття класу, об'єкта. Стан і поведінка об'єкта. Інкапсуляція.
4. Структури у порівнянні з класами. Доступ по замовчуванню.
5. Визначення класу. Поняття поля, методу, повідомлення.
6. Класи та об'єкти. Протокол опису класу. Створення та ініціалізація об'єктів. Доступ до полів та методів.
7. Функції – елементи класу. Перевантаження функцій - елементів класу.
8. Конструктори. Правила визначення. Використання списку ініціалізації у конструкторі. Приклад.
9. Перевантажені конструктори. Деструктор.
10. Поверхнєве та глибинне копіювання. Конструктор копії.
11. Конструктор перетворення. Відмова від використання конструктора у неявних перетвореннях типу.
12. Вказівник `this`. Явне та неявне використання.
13. Константні функції елементи. Константні об'єкти. Використання специфікатора `mutable`. Приклади.
14. Вказівники на об'єкти. Динамічні об'єкти та масиви об'єктів.
15. Поля екземпляру та статичні поля класу. Статичні та функції класу.
16. Дружні функції класу. Дружні класи.
17. Перевантаження операцій. Загальна концепція.
18. Глобальне перевантаження операцій. Правила.
19. Перевантаження операцій у межах класу. Правила.
20. Перевантаження операцій зчитування з потоку вводу та вставки у потік виводу.
21. Вкладені об'єкти. Виклик конструктора для вкладеного об'єкта. Агрегація об'єктів. Композиція об'єктів.
22. Механізм успадкування. Керування доступом при успадкуванні.
23. Успадкування. Виклики конструкторів та деструкторів при спадкуванні класів.
24. Множинне успадкування. Виклики конструкторів для базових класів.
25. Множинне успадкування. Конфлікти. Проблема ромба. Віртуальний базовий клас.
26. Поліморфізм. Статичний та динамічний поліморфізм.
27. Раннє та пізнє зв'язування. Віртуальні функції.
28. Чисті віртуальні функції та абстрактні класи.
29. Поняття інтерфейсу. Використання.
30. Механізм обробки винятків. Створення та використання власних типів винятків.
31. Робота із стандартними винятками.
32. Шаблони класів. Стандартні шаблонні типи `pair<>`, `tuple<>`, `vector<>`.
33. Бібліотека STL. Поняття контейнера, алгоритму, ітератора.
34. Використання контейнерів STL: вектор, список, дек, черга, стек.
35. Використання контейнерів STL: множина, карта.
36. Алгоритми роботи з контейнерами.
37. Ітератори STL. Застосування.
38. Файлові потоки. Режими відкриття файлу. Запис та зчитування.

18. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Бичков О.С. Основи сучасного програмування: Підручник. К. : ВПЦ «Київський університет», 2008. 272 с.
2. Белов Ю. А., Карнаух Т. О., Коваль Ю. В., Ставровський А. Б. Вступ до програмування мовою C++. Організація обчислень. К. : ВПЦ «Київський університет», 2012. 175 с.
3. Кравець П.О. Об'єктно-орієнтоване програмування: Навч. посібник. Львів : Видавництво Львівської політехніки, 2012. 624 с.
4. Stephen Prata. C++ Primer Plus : 6th Edition. Boston : Addison-Wesley Professional, 2013. 1080 p.
5. Bjarne Stroustrup. The C++ Programming Language. Boston : Addison-Wesley Professional, 2013. 1361 p.
6. Paul Deitel, Harvey Deitel. C++ How to Program, 10 edition. NY: Pearson: 2016. 1080 p.
7. Tony Gaddis. Starting Out with C++ from Control Structures to Objects. NY: Pearson, 2017. 1344 p.

Допоміжна:

9. Ivor Horton . Beginning C++20: From Novice to Professional 6th Edition. NY: Apress, 2020. 825 p.
10. Paul Deitel, Harvey Deitel. C++20 for Programmers: An Objects-Natural Approach : 3rd Edition. NJ: Pearson, 2022. 960 p.

19. ІНФОРМАЦІЙНІ (ІНТЕРНЕТ) РЕСУРСИ

- The Standard. URL : <https://isocpp.org/std/the-standard>
- Уроки програмування на C++. URL : <https://acode.com.ua/uroki-po-cpp/>
- C++ Language Reference URL : <https://learn.microsoft.com/en-us/cpp/cpp/cpp-language-reference?view=msvc-170&viewFallbackFrom=vs-2022>
- Learn.cpp. URL: <https://www.learncpp.com/>
- C++ Tutorial for Complete Beginners URL: <https://www.udemy.com/course/free-learn-c-tutorial-beginners/>
- C++ Tutorial. URL: <https://www.w3schools.com/cpp/default.asp>
- Learn Object Oriented Programming OOP in C++. URL: <https://www.udemy.com/course/c-programming-oops-concepts/>
- C++ OOP : Object Oriented Programming for C++ Projects. URL : <https://www.udemy.com/course/object-oriented-c-plus-plus-programming/>
- Complete Modern C++ (C++11/14/17). URL: <https://www.udemy.com/course/beg-modern-cpp/>