

Центральноукраїнський національний технічний університет
Кафедра кібербезпеки та програмного забезпечення



“ЗАТВЕРДЖУЮ”

Проректор з науково-педагогічної роботи

Андрій КИРИЧЕНКО

“ 15 ” 08 2025 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Об’єктно-орієнтоване програмування

(назва навчальної дисципліни)

спеціальність F7 / 123 Комп’ютерна інженерія

(шифр і назва спеціальності)

спеціалізація «Комп’ютерна інженерія»

(назва освітньої програми)

факультет механіко-технологічний

(назва факультету)

2025-2026 навчальний рік

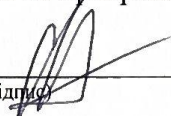
Розробники: старший викладач кафедри кібербезпеки та програмного забезпечення, доктор філософії Усік П.С., senior software engineer Broadway Gaming Ltd. Михайлюк О.В., викладач кафедри кібербезпеки та програмного забезпечення, Козірова Н. Л.

(вказати авторів, їхні посади, наукові ступені та вчені звання)

Робочу програму схвалено на засіданні кафедри кібербезпеки та програмного забезпечення


Протокол від №15 від 26 червня 2025 року

Завідувач кафедри кібербезпеки та програмного забезпечення _____



(Олексій СМІРНОВ)
(прізвище та ініціали)

Декан факультету механіко-технологічний



(Віталій МАЖАРА)
(прізвище та ініціали)

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів ЄКТС – 4	Галузь знань <u>12/F Інформаційні технології</u> (шифр і назва)	Спеціальної (фахової) підготовки	
Загальна кількість годин: денна форма навчання – 120 заочна форма навчання – 120	Спеціальність: <u>123/F7 «Комп'ютерна інженерія»</u>	Рік підготовки	
		2-й	
		Семестр	
		3-й	
Тижневих годин для денної форми навчання: 3-й семестр аудиторних – 3 год. самостійної роботи студента – 4,5 год.	Освітньо-кваліфікаційний рівень: Бакалавр	Лекції	
		16	2
		Лабораторні	
		32	4
		Самостійна робота	
		72	114
		Вид контролю: зал.	

Мова навчання українська

2. Мета та завдання навчальної дисципліни

Метою викладання дисципліни «Об'єктно-орієнтоване програмування» є отримання студентами знання, уміння і навичок для побудови алгоритмічних методик вирішення складних задач на етапах об'єктно-орієнтованого аналізу (ООА) та об'єктно-орієнтованого проектування (ООД), проектування програм на мові С++ в рамках процедурної та об'єктно-орієнтованої парадигм програмування. Основними завданнями вивчення дисципліни є формування наступних **компетенцій бакалавра з комп'ютерної інженерії:**

– Р2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

– Р3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

У результаті вивчення дисципліни студент повинен забезпечити наступні **програмні результати навчання:**

– N6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей;

– N20. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

Передумови для вивчення дисципліни (структурно-логічна схема підготовки фахівця).

Враховуючи послідовність накопичення знань та інформації, дисципліна вивчається після викладання наступних дисциплін: «Базові методології та технології програмування», «Основи комп'ютерних технологій».

Результати навчання:

– знати новітні технології в галузі комп'ютерної інженерії. Використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення. Визначати критерії, яким повинен задовольняти проект, щоб його легко було супроводжувати і модифікувати. Створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж. Використовувати класи-шаблони стандартної бібліотеки мови С ++ (STL) та узагальнені алгоритми при написанні програм.) та узагальнені алгоритми при написанні програм. Застосовувати шаблони (патерни)..

– вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей. Розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності. Адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

3. Програма навчальної дисципліни

3 семестр

Тема 1. Основні поняття ООП. Класи та об'єкти. Функції доступу. Вказівник `this`. Конструктори і деструктори.

Тема 2. Статичні змінні та методи. Типи зв'язків між класами.

Тема 3. Наслідування.

Тема 4. Поліморфізм. Обробка винятків.

Тема 5. Перевантаження операторів.

Тема 6. Шаблони в C++.

Тема 7. Контейнерні класи. Стандартна бібліотека шаблонів (STL) в C++. (Послідовні контейнери, асоціативні контейнери)

Тема 8. Контейнерні класи. Стандартна бібліотека шаблонів (STL) в C++. (Контейнери-адаптери, ітератори)

4. Структура навчальної дисципліни

Назви тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб.	інд.	с. р.		л	п	лаб.	інд.	с. р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Тема 1. Основні поняття ООП. Класи та об'єкти. Функції доступу. Вказівник this. Конструктори і деструктори.	15	2		4		9	15	0,25		0,5		14,25
Тема 2. Статичні змінні та методи. Типи зв'язків між класами.	15	2		4		9	15	0,25		0,5		14,25
Тема 3. Наслідування.	15	2		4		9	15	0,25		0,5		14,25
Тема 4. Поліморфізм. Обробка винятків.	15	2		4		9	15	0,25		0,5		14,25
Тема 5. Перевантаження операторів.	15	2		4		9	15	0,25		0,5		14,25
Тема 6. Шаблони в C++.	15	2		4		9	15	0,25		0,5		14,25
Тема 7. Контейнерні класи. Стандартна бібліотека шаблонів (STL) в C++. (Послідовні контейнери, асоціативні контейнери).	15	2		4		9	15	0,25		0,5		14,25
Тема 8. Контейнерні класи. Стандартна бібліотека шаблонів (STL) в C++. (Контейнери-адаптери, ітератори)	15	2		4		9	15	0,25		0,5		14,25
Всього, год.	120	16		32		72	120	2		4		114

5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин	
		денна форма навчання	заочна форма навчання
3 семестр			
1	Основні поняття ООП. Класи та об'єкти. Функції доступу. Вказівник this. Конструктори і деструктори.	4	0,5
2	Типи зв'язків між класами: асоціація, агрегація, композиція, реалізація та залежність.	4	0,5
3	Наслідування. Специфікатори доступу.	4	0,5
4	Поліморфізм. Обробка винятків.	4	0,5
5	Перевантаження операторів.	4	0,5
6	Шаблони в C++.	4	0,5
7	Контейнерні класи. Стандартна бібліотека шаблонів (STL) в C++. (Послідовні контейнери, асоціативні контейнери)	4	0,5
8	Контейнерні класи. Стандартна бібліотека шаблонів (STL) в C++. (Контейнери-адаптери, ітератори)	4	0,5
Усього годин		32	4

6. Самостійна робота

№ з/п	Назва теми	Кількість годин	
		денна форма навчання	заочна форма навчання
3 семестр			
1	Тема 1. Основні поняття ООП. Класи та об'єкти. Функції доступу. Вказівник this. Конструктори і деструктори.	9	14,25
2	Тема 2. Статичні змінні та методи. Типи зв'язків між класами.	9	14,25
3	Тема 3. Наслідування.	9	14,25
4	Тема 4. Поліморфізм. Обробка винятків.	9	14,25
5	Тема 5. Перевантаження операторів.	9	14,25
6	Тема 6. Шаблони в C++.	9	14,25
7	Тема 7. Контейнерні класи. Стандартна бібліотека шаблонів (STL) в C++. (Послідовні контейнери, асоціативні контейнери)	9	14,25
8	Тема 8. Контейнерні класи. Стандартна бібліотека шаблонів (STL) в C++. (Контейнери-адаптери, ітератори)	9	14,25
Усього годин		72	114

Для опанування матеріалу дисципліни «Об'єктно-орієнтованого програмування» окрім лекційних та лабораторних занять, тобто аудиторного навантаження, значна увага приділяється самостійній роботі.

До основних видів самостійної роботи студента відносимо:

1. Вивчення лекційного матеріалу.
2. Робота з літературними джерелами.
3. Розв'язання практичних задач.
4. Підготовка до модульних, підсумкового контролю, екзамену (денна) та заліку (заочна).
5. Виконання контрольної роботи для заочної форми навчання.

Студенти заочної форми навчання (ЗФН) здебільшого вивчають матеріал самостійно впродовж семестру, тобто самостійно відпрацьовують теми лекцій, а також лабораторних робіт. Для них на початку семестру проводиться установча сесія, під час якої начитують лекції та проводять лабораторні роботи.

7. Індивідуальні завдання

Індивідуальне завдання є формою організації навчальної діяльності, що забезпечує поглиблене опанування принципів об'єктно-орієнтованого програмування та сприяє розвитку творчих і проектних здібностей студентів. Виконання такого завдання передбачає самостійну або командну (до 3 осіб) роботу над програмним проектом, орієнтованим на створення гри з використанням основних механізмів ООП.

Індивідуальне завдання є об'ємною роботою, яка не входить до тижневого аудиторного навантаження та виконується студентами у позааудиторний час. Контроль за виконанням здійснюється викладачем поетапно відповідно до

встановленого графіка, що передбачає перевірку проміжних результатів. Після завершення роботи студенти презентують отримані результати у вигляді демонстрації проекту.

Виконання індивідуального завдання сприяє формуванню практичних навичок програмування, умінню застосовувати теоретичні положення ООП на практиці та розвитку навичок командної взаємодії. Результати виконання індивідуального завдання можуть враховуватися як додаткові бали до підсумкової оцінки.

7. Методи навчання і викладання

У дисципліні «Об'єктно-орієнтоване програмування» використовуються комбіновані методи навчання, що поєднують теоретичний матеріал і практичні навички.

Словесні методи: пояснення теоретичного матеріалу, ознайомлення з основними поняттями та принципами об'єктно-орієнтованого підходу, обговорення ключових ідей та відповідь на запитання з метою уточнення та закріплення матеріалу.

Наочні методи: використання мультимедійних презентацій, схематичне зображення структури програм і взаємодії їх елементів, демонстрація прикладів коду для ілюстрації роботи об'єктів, класів та їх взаємозв'язків.

Практичні методи: виконання лабораторних робіт під керівництвом викладача, що передбачають створення програмних рішень із використанням принципів ООП, застосування базових і розширених механізмів роботи з класами, а також роботу з файлами та стандартними інструментами мови програмування.

9. Критерії й засоби оцінювання

Види контролю: поточний, підсумковий.

Методи контролю: спостереження за навчальною діяльністю студентів, усне опитування, захист результатів виконання лабораторних робіт.

Форма підсумкового контролю: семестр 3 (осінній) – залік.

Оцінювання результатів навчання здійснюється через поточний і підсумковий контроль, кожен з яких має свої особливості та критерії.

Поточний контроль передбачає оцінювання кожної лабораторної роботи. Основними критеріями є правильність виконання завдання, якість захисту роботи та дотримання встановлених термінів. У процесі оцінювання враховується рівень розуміння теоретичного матеріалу та сформованість практичних навичок. Особлива увага приділяється розумінню концепції ООП та вмінню застосовувати її під час розроблення програмних рішень.

За бажанням студент може виконати індивідуальне, яке дає можливість поглибити рівень знань та позитивно вплинути на підсумкову оцінку.

Підсумковий контроль здійснюється у формі заліку, який оцінює ступінь засвоєння теоретичних положень дисципліни та здатність студента застосовувати отримані знання на практиці. Однак, в процесі вивчення дисципліни передбачено

виконання комплексу робіт під час лекційних та лабораторних занять, а також індивідуальних завдань. У зв'язку з цим здобувачам вищої освіти може бути виставлена оцінка понад 60 балів без обов'язкового виконання підсумкової залікової роботи.

Критерії оцінювання роботи на лекціях

Передбачено 8 лекційних занять, за кожне з яких студент може отримати до 5 балів.

Бали нараховуються за такі види діяльності:

- підготовка теоретичного матеріалу (опрацювання теми) - 1 бал;
- активність та участь в обговореннях - 1 бал;
- відповіді на теоретичні питання (правильність, повнота) - 1 бал;
- аналіз і порівняння підходів або рішень (зокрема різних реалізацій) - 1 бал;
- виявлення та пояснення помилок у запропонованих прикладах - 1 бал.

Максимальна кількість балів за одне заняття - 5 балів.

Критерії оцінювання лабораторних робіт

Передбачено 8 лабораторних робіт, за кожен з яких студент може отримати до 8 балів, крім лабораторної роботи №1, максимальна оцінка за яку становить 6 балів.

Оцінювання здійснюється за такими рівнями:

7-8 балів (відмінно) - завдання виконано повністю, результати коректні, студент демонструє глибоке розуміння матеріалу та впевнено захищає роботу;

5-6 балів (добре) - завдання виконано в повному обсязі, але наявні незначні помилки або неточності;

3-4 бали (задовільно) - завдання виконано частково, спостерігається поверхнєве розуміння матеріалу;

1-2 бали (мінімально) - робота виконана формально, студент демонструє слабе розуміння;

0 балів - робота не виконана або не захищена.

Для лабораторної роботи №1 оцінювання здійснюється пропорційно до максимальної кількості балів.

Розподіл балів, які отримують студенти при вивченні дисципліни «Об'єктно-орієнтоване програмування»

Змістовий модуль 1								Змістовий модуль 2								Сума
Т1		Т2		Т3		Т4		Т5		Т6		Т7		Т8		
Л1	ЛР1	Л2	ЛР2	Л3	ЛР3	Л4	ЛР4	Л5	ЛР5	Л6	ЛР6	Л7	ЛР7	Л8	ЛР8	
5	6	5	8	5	8	5	8	5	8	5	6	5	8	5	8	
50								50								100

Примітка: Т1, Т2,...,Т14 – тема, Л – теоретичні (лекційні) заняття, ЛР – лабораторні заняття

При виставленні оцінки враховуються результати навчальної роботи студента протягом семестру

Критерії оцінки заліку:

– «зараховано» – студент має стійкі знання про основні поняття дисципліни, може сформулювати взаємозв'язки між поняттями.

– «незараховано» – студент має значні пропуски в знаннях, не може сформулювати взаємозв'язку між поняттями, що вивчаються в курсі, не має уявлення про більшість основних понять дисципліни, що вивчається.

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
1-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

9. Рекомендована література

Базова

1. Григорович В. Г. Об'єктно-орієнтоване програмування : навч. посіб. Ч. 1. Основні поняття. Класи та об'єкти. Конструктори та перевантаження операцій. Успадкування. Віртуальні функції. Поліморфізм. – Львів : Видавництво Магнолія 2006, 2025. – 284 с. (бібліотека ЦНТУ)
2. Григорович В. Г. Об'єктно-орієнтоване програмування : навч. посіб. Ч. 2. Винятки. Контейнери та шаблони STL - стандартна бібліотека шаблонів S.O.L.L.D. – Львів : Видавництво Магнолія 2006, 2025. – 343 с.(бібліотека ЦНТУ)
3. Зеленський О.С., Лисенко В.С. ОБ'ЄКТНО-ОРИЄНТОВАНЕ ПРОГРАМУВАННЯ на C++ : навч. посіб. – Кривий Ріг: Державний університет економіки і технологій, 2023. – 215 с.
4. B. Stroustrup: The C++ Programming Language (Fourth Edition). May 2013. Addison Wesley. Reading Mass. USA. May 2013. ISBN 0-321-56384-0. 1360 pages. Softcover, hardcover, and electronic versions.
5. Ira Pohl: Object-oriented programming using C++. 1997. Addison-Wesley. ISBN 978-0201895506. 543 pages.

6. B. Stroustrup: Programming -- Principles and Practice Using C++. December 2008. Addison-Wesley. ISBN 978-0321543721. 1264 pages. Softcover.
7. B. Stroustrup: Programming -- Principles and Practice Using C++ (Second Edition). May 2014. Addison-Wesley. ISBN 978-0321992789. 1312 pages. Softcover and electronic versions.
8. B. Stroustrup: A Tour of C++ (Second Edition). July 2018. Addison-Wesley. ISBN 978-0-13-499783-4. 240 pages. Softcover and electronic versions.
9. Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Conallen, Kelli A. Houston / Object-Oriented Analysis and Design with Applications (3rd Edition) - 2007/ 720 p.
10. Rainer Grimm. C++ Core Guidelines. Addison-Wesley Professional. 2022. 403 с.
11. Bill Weinman. C++20 STL Cookbook. Packt Publishing. 2022. 450 с.

Допоміжна

12. Жуковський С.С., Вакалюк Т.А. Програмування мовою C++. Структурне програмування (лабораторний практикум). Навчальний посібник для студентів фізико-математичного факультету. – Житомир: Вид-во ЖДУ, 2011. – 92 с. (видання друге, перероблене та доповнене).
13. Грицюк Ю.І., Рак Т.Є. Г 85 Об'єктно-орієнтоване програмування мовою C++ : навчальний посібник. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 404 с. – Статистика: іл. 18, табл. 12, бібліогр. 34. ISBN 978-966-3466-86-3BN 978-966-3466-86-3
14. Adrian Ostrowski, Piotr Gaczkowski. Software Architecture with C++. Packt Publishing. 2021. 522 с.

Методичне забезпечення

15. Усік П. С., Козірова Н.Л., Ткачук Р.О. «Об'єктно-орієнтоване програмування», Методичні вказівки до виконання лабораторних робіт для студентів за спеціальностями 123 “Комп’ютерна інженерія”, 122 “Комп’ютерні науки”, 125 “Кібербезпека”/ М-во освіти і науки України, Центральн. нац. техн. ун-т; [уклад. : П. С. Усік, Н. Л. Козірова, Р. О. Ткачук] – Кропивницький: ЦНТУ, 2024. – 114 с. Режим доступу: <https://dspace.kntu.kr.ua/items/ec70430c-2ba4-4520-b905-86fcfcc9720d>

Інформаційні ресурси

16. Курс «Об'єктно-орієнтоване програмування» на сервері дистанційної освіти ЦНТУ. – URL: <https://moodle.kntu.kr.ua/course/view.php?id=1046>
17. Онлайн-курси Prometheus. – URL: <https://prometheus.org.ua/>
18. Онлайн-курси Coursera. – URL: <https://www.coursera.org>
19. Академія Cisco. – URL: <https://www.netacad.com>
20. Он-лайн ресурс з інформаційних технологій. – URL: <https://dou.ua/>
21. Пошукова система. – URL: <https://www.google.com/>
22. Он-лайн ресурс перегляду відеоуроків. – URL: <https://www.youtube.com>
23. Он-лайн ресурс LEARN C++. – URL: <https://www.learncpp.com/>
24. Он-лайн ресурс aCode. – URL: <https://acode.com.ua/>

