

Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра комп'ютерних наук та інформаційних технологій (№ 302)

ЗАТВЕРДЖУЮ

Гарант освітньої програми

 Олексій ГУБКА
(підпис) (ініціали та прізвище)

« 26 » 08 2026 р.

**СИЛАБУС ОBOB'ЯЗKОВОЇ
HABЧАЛЬНОЇ ДИСЦИПЛІНИ**

«Кросплатформне програмування»

(назва навчальної дисципліни)

Галузь знань:

ґ «Інформаційні технології»

(шифр і найменування галузі знань)

Спеціальність:

ґ3 «Комп'ютерні науки»

(код і найменування спеціальності)

Освітня програма:

«Віртуальні та інтелектуальні технології програмування»

(найменування освітньої програми)

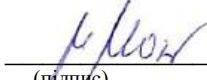
Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Вводиться в дію з «01» вересня 2026 р.

Харків 2026

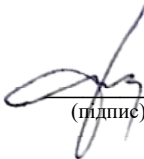
Розробник: Мирослав МОМОТ, доцент, к.т.н., доцент
(прізвище та ініціали, посада, наукова ступінь та вчене звання)


(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри (№ 302)
комп'ютерних наук та інформаційних технологій
(назва кафедри)

Протокол № 694/07 від « 26 » 08 2026 р.

В.о. завідувача кафедри Д.Т.Н., проф.
(науковий ступінь і вчене звання)


(підпис)

Олег ФЕДОРОВИЧ
(ім'я та прізвище)

Погоджено з представником здобувачів освіти:

здобувачка освіти групи 336


(підпис)

Марія ШЕВЧЕНКО
(ім'я та прізвище)

1. Загальна інформація про викладача



Момот Мирослав Олександрович, к.т.н., доцент, доцент кафедри комп'ютерних наук та інформаційних технологій (№ 302). З 2001 з року викладає в університеті дисципліни, пов'язані з програмуванням, в даний час наступні:

- «Кросплатформне програмування»;
- «Сучасні технології програмування»;
- «Технології системного аналізу»;
- «Технологія створення програмних продуктів»;
- «Управління створенням програмних продуктів».

Напрями наукових досліджень: інформаційне моделювання при створенні й модернізації мереж зв'язку та передачі даних, інформаційне моделювання телекомунікаційних послуг, системи верифікації і моделювання технологічного процесу складання.

2. Опис навчальної дисципліни

Семестр, в якому викладається дисципліна – 3 семестр.

Обсяг дисципліни: 7 кредитів ЄКТС (210 годин), у тому числі аудиторних – 80 годин, самостійної роботи здобувачів – 130 годин.

Форми здобуття освіти - денна, дистанційна.

Дисципліна – обов'язкова.

Види навчальної діяльності – лекції, лабораторні заняття, самостійна робота здобувача.

Види контролю – поточний, модульний та підсумковий (семестровий) контроль (іспит).

Мова викладання – українська.

Пререквізити:

ОК2. Основи програмування

ОК3. Моделі та методи дискретної математики

ОК4. Вступ до спеціальності

ОК10. Створення візуальних інтерфейсів

ОК11. Структури даних та їх віртуалізація

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета – дати знання про методи, принципи та концептуальні положення об'єктно-орієнтованого програмування для створення сучасного кроссплатформного програмного забезпечення.

Завдання – навчити студентів використовувати в практичній діяльності можливості об'єктно-орієнтованого програмування при створенні кроссплатформного програмного забезпечення.

Компетентності, які набуваються:

Інтегральна компетентність:

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

Загальні компетентності:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК5. Здатність спілкуватися іноземною мовою.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК11. Здатність приймати обґрунтовані рішення.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

Спеціальні (фахові, предметні) компетентності:

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проєктування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК8. Здатність проєктувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК12. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.

СК16. Здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації.

СК17. Здатність до розробки програмного забезпечення, з урахуванням віртуалізації ІТ-інфраструктури, для завдань інтелектуального управління об'єктами та процесами у реальному часі.

Програмні результати навчання:

ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПР10. Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.

ПР11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проєктну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

ПР13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

ПР14. Застосовувати методи розробки програмного забезпечення для створення систем управління та мережевих систем, які використовуються в критичних галузях та процесах.

ПР15. Застосовувати знання методології та CASE-засобів проєктування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проєктування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

ПР18. Виконувати розробку інструментальних засобів та програмного забезпечення для управління складними системами та процесами у реальному часі.

4. Зміст навчальної дисципліни

Модуль 1.

Змістовний модуль 1.

Тема 1. Вступ до навчальної дисципліни «Кросплатформне програмування». Передумови появи об'єктно-орієнтованого програмування (ООП) для розробки складного програмного забезпечення

- Тема та питання лекції:

Предмет, задачі та структура курсу. Зв'язок курсу з іншими дисциплінами спеціальності. Список рекомендованої літератури.

Складність, властива програмному забезпеченню. Організована та неорганізована складність. Канонічна форма складної програмної системи. Алгоритмічна та об'єктно-орієнтована декомпозиція. Роль декомпозиції, абстракції та ієрархії при розробці програмного забезпечення.

ООП як фундамент, який дозволяє створювати гнучкі кросплатформні додатки. Покоління мов програмування. Топологія мов різного покоління. Структурна та процедурна парадигми програмування. Об'єктно-орієнтована парадигма програмування. Класифікація об'єктно-орієнтованих мов програмування.

Організована та неорганізована складність. Канонічна форма складної програмної системи. Структурна та процедурна парадигми програмування.

- *Самостійна робота здобувача освіти:*

Опрацювання матеріалу лекцій. Формування питань до викладача. Вивчення термінів та визначень.

Тема 2. Структури та класи в C/C++/C#, Object Pascal/Delphi, Python.

- *Тема та питання лекції:*

Класи. Структури як класи. Характерні особливості, відмінності структур в C, C++, C#, а також записів в Object Pascal/Delphi. Виділення та визволення пам'яті. Особливості об'яви конструкторів та деструкторів та перевантаження конструкторів в C++, Object Pascal/Delphi, Python. Функції-члени та дані-члени.

- *Лабораторна робота: «Використання структур у C++/C# та записів в Object Pascal/Delphi».*

- *Самостійна робота здобувача освіти:*

Опрацювання матеріалу лекцій. Підготовка до виконання лабораторної роботи. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Тема 3. Інтерфейси та реалізації класів в C++/C#, Object Pascal/Delphi, Python. Області видимості.

- *Тема та питання лекції:*

Інтерфейси та реалізації класів в C++. Області видимості. Показчик this. Приховання даних. Розділи класів. Друзі класу. Статичні члени. Вкладені класи.

- *Лабораторна робота: «Використання класів в C++/C# та Delphi. Конструювання і знищення об'єктів. Динамічне виділення та визволення пам'яті».*

- *Самостійна робота здобувача освіти:*

Опрацювання матеріалу лекцій. Підготовка до виконання лабораторної роботи. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Тема 4. Перевантаження операцій у мові C++/C#, Object Pascal/Delphi, Python.

- *Тема та питання лекції:*

Бінарні й унарні операції. Наперед визначені значення операцій. Визначене перетворення типу. Обмеження на перевантаження операцій, особливості у мовах C++/C#, Object Pascal/Delphi, Python.

- *Лабораторна робота: «Перевантаження операцій у мові C++/C#».*

- *Самостійна робота здобувача освіти:*

Опрацювання матеріалу лекцій. Підготовка до виконання лабораторної роботи. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Модульний контроль 1

Змістовний модуль 2.

Тема 5. Похідні класи. Побудова ієрархій класів та поліморфізм.

- Тема та питання лекції:

Особливості об'яви похідних класів. Одиночне та множинне спадкування. Поля типу. Дерево та граф спадкування ієрархій класів. Віртуальні функції. Поліморфна поведінка класів, які зв'язані спадкуванням. Чисті віртуальні функції. Абстрактний базовий клас.

- Лабораторна робота: «Вивчення спадкування в ООП».

- Самостійна робота здобувача освіти:

Опрацювання матеріалу лекцій. Підготовка до виконання лабораторної роботи. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Тема 6. Обробка виняткових ситуацій.

- Тема та питання лекції:

Обробка помилок в стандартному C. Поняття винятку, виняткової ситуації стосовно програмного забезпечення. Оператори try, catch, throw. Вкладення блоків try-catch. Перехоплення виняткової ситуації. Оброблювач виняткової ситуації. Повторна генерація виняткової ситуації. Особливості обробки виняткових ситуацій у мовах C++/C#, Object Pascal/Delphi, Python.

- Лабораторна робота: «Обробка виняткових ситуацій в C++/C#».

- Самостійна робота здобувача освіти:

Опрацювання матеріалу лекцій. Підготовка до виконання лабораторної роботи. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Тема 7. Шаблони функцій та класів.

- Тема та питання лекції:

Шаблони функцій, їх переваги та недоліки у порівнянні з макросами. Об'ява та виклик шаблонної функції. Шаблони класів. Об'ява та використання шаблонів класів. Параметри шаблонів. Інстаціювання. Нетипові параметри шаблонів. Особливості узагальнень у мові C#.

Огляд шаблонів стандартної бібліотеки STL (Standard Template Library) в C++, а також узагальнених колекцій C# (System.Collections.Generic). Контейнерні типи, колекції, ітератори, черги, стеки.

- Лабораторна робота: «Обробка виняткових ситуацій в C++/C#».

- Самостійна робота здобувача освіти:

Опрацювання матеріалу лекцій. Підготовка до виконання лабораторної роботи. Формування питань до викладача. Вивчення термінів та визначень. Оформлення лабораторної роботи та підготовка до її здачі.

Тема 8. Об'єктний підхід. Об'єктно-орієнтовані аналіз, проектування та програмування.

- Тема та питання лекції:

Складові об'єктного підходу: абстрагування, інкапсуляція, модульність, ієрархія, слабка та сильна типізація, паралелізм, зберегаємість. Поняття об'єктно-орієнтованого аналізу (ООА), проектування (OOD) та програмування (OOP), їх взаємозв'язок.

Об'єкт як основа ООП. Стан та поведінка об'єкта, ідентичність об'єктів. Зв'язок між об'єктами – передача повідомлень. Ролі об'єктів: актор, сервер та агент. Класи. Відносини між класами. Різні розуміння поняття інтерфейсу, розділи інтерфейсу класу. Взаємозв'язок класів та об'єктів.

Асоціація як семантичний зв'язок між класами. Потужність зв'язку. Спадкування. Одиночне та множинне спадкування. Графічне позначення асоціації та спадкування. Приклади асоціації та спадкування. Використання. Клієнт та сервер при використанні. Агрегація. Фізичне включення як складова форма агрегації. Графічне позначення агрегації та використання. Приклади агрегації та використання. Інстанцування при використанні шаблонів. Графічне позначення інстанцування та метакласів. Приклади інстанцування та метакласів. Поняття класифікації як виділення класів та об'єктів предметної області. Проблеми класифікації.

Якість програмного продукту. Класичні міри якості програмного продукту. Залежність кількості знайдених помилок від часу. Щільність помилок. Якість класів та об'єктів. Об'єктно-орієнтовані міри якості програмного продукту. Зважена насиченість класу методами. Глибина дерева спадку. Число нащадків. Зачеплення класів. Відклик на клас. Нестача зв'язності у методах

Інструменти розробки програмного забезпечення. Набір інструментів попередніх поколінь мов програмування – редактор, компілятор, компоновальник, завантажувач. Набір інструментів об'єктно-орієнтованих мов програмування – графічний редактор, браузер класів, інкрементний компілятор, налагоджувач, компоновальник, завантажувач, інструменти управління конфігурацією та контролю версій проекту ПЗ, генератор графічного інтерфейсу користувача.

Перспективи розвитку об'єктно-орієнтованих аналізу, проектування та програмування.

- Самостійна робота здобувача освіти:

Опрацювання матеріалу лекцій. Формування питань до викладача. Вивчення термінів та визначень.

Модульний контроль 2

Модуль 2.

Індивідуальне завдання (РР).

- Форма занять: самостійна робота.

5. Індивідуальні завдання

Виконання РР за варіантами на тему «Побудова діаграм взаємодії класів і об'єктів».

6. Методи навчання

Проведення аудиторних лекцій, лабораторних занять, індивідуальні консультації (при необхідності), самостійна робота студентів за матеріалами, опублікованими кафедрою (методичні посібники).

7. Методи контролю

Проведення поточного контролю, письмового модульного контролю, підсумковий контроль у вигляді іспиту.

8. Критерії оцінювання та розподіл балів, які отримують здобувачі

Таблиця 8.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Виконання і захист лабораторних робіт	0...10	3	0...30
Модульний контроль	0...15	1	0...15
Змістовний модуль 2			
Виконання і захист лабораторних робіт	0...10	3	0...30
Модульний контроль	0...15	1	0...15
Виконання і захист РР	0...10	1	0...10
Усього за семестр			0...100

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Білет для іспиту складається з трьох запитань: двох теоретичних (максимальна кількість балів за повну та правильну відповідь на одне запитання - 30) та одного практичного (максимальна кількість балів - 40). Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.

Таблиця 8.2 – Шкали оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційний залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

Критерії оцінювання роботи здобувача протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Відпрацювати та захистити всі лабораторні роботи та РР. Вміти застосовувати основи об'єктно-орієнтованого аналізу, знати основні положення об'єктно-орієнтованого програмування, методи, способи та засоби розробки програм у рамках цього напрямку. Вміти писати програмний код структур та класів.

Добре (75 - 89). Твердо знати мінімум знань, виконати усі завдання. Показати вміння виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з обґрунтуванням рішень та заходів, які запропоновано у роботах. Вміти робити об'єктно-орієнтований аналіз предметної області, робити декомпозицію задачі у вигляді окремих класів та об'єктів, а також взаємозв'язків між ними. Вміти пояснювати складні алгоритми, які реалізовані в програмах.

Відмінно (90 - 100). Повно знати основний та додатковий матеріал. Знати усі теми. Орієнтуватися у підручниках та посібниках. Досконально знати складові об'єктно-орієнтованого програмування, які використовуються при розробці складних програм. Вміти реалізовувати спроектовані ієрархії класів у вигляді закінченого програмного забезпечення; знати сучасні методи та засоби розробки алгоритмів та програм, способи запису алгоритму мовою високого рівня, способи налагодження, випробування та документування програм. Безпомилково виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з докладним обґрунтуванням рішень та заходів, які запропоновано у роботах.

9. Політика навчального курсу

Відвідування занять. Здобувачі освіти, які за певних обставин не можуть відвідувати практичні заняття регулярно, повинні протягом тижня узгодити із викладачем графік індивідуального відпрацювання пропущених занять. Окремі пропущені заняття мають бути відпрацьовані на найближчій консультації протягом тижня після їх пропуску.

Дотримання вимог академічної доброчесності здобувачами освіти під час вивчення навчальної дисципліни. Під час вивчення навчальної дисципліни здобувачі освіти мають дотримуватися загальноприйнятих морально-етичних норм і правил поведінки, вимог академічної доброчесності, передбачених Кодексом академічної доброчесності та забезпечення її дотримання Національного аерокосмічного університету «Харківський авіаційний інститут».

Вирішення конфліктів. Порядок і процедури врегулювання конфліктів, пов'язаних із корупційними діями, зіткненням інтересів, різними формами дискримінації, сексуальними домаганнями, міжособистісними стосунками та іншими ситуаціями, що можуть виникнути під час навчання, а також правила етичної поведінки регламентуються Кодексом етики в Національному аерокосмічному університеті «Харківський авіаційний інститут».

10. Методичне забезпечення

1. Електронні матеріали «Кросплатформне програмування» [Електронний ресурс] – Режим доступу: <https://mentor.khai.edu/course/view.php?id=1835>

11. Рекомендована література

Базова

1. Бублик В.В. Об'єктно-орієнтоване програмування: – К.: ІТкнига, 2015. – 624 с.
2. Alan Dennis, David Paul Tegarden, Barbara Haley Wixom. Systems Analysis and Design: An Object-Oriented Approach with UML– Wiley, 2015. – 544 p.
3. Andrew Troelsen. Pro C# 7: With .NET and .NET Core. - 8th ed. - Apress, 2017. - 1437 p.
4. Bjarne Stroustrup. Programming: Principles and Practice Using C++: 2nd edition. – Addison-Wesley Professional, 2014 –1312 p.
5. Don Hermes. Xamarin Mobile Application Development: Cross-Platform C# and Xamarin.Forms Fundamentals. – Berkeley : Apress, 2015. – 432 p.

Допоміжна

1. І.М.Дудзяний. Об'єктно-орієнтоване моделювання програмних систем: Навчальний посібник. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. - 108 с.
2. Grady Booch. Object-Oriented Analysis and Design with Applications. – Addison-Wesley Professional, 2011. – 608 p.

12. Інформаційні ресурси

1. Об'єктно орієнтоване програмування (C#). [Електронний ресурс] – Режим доступу: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/object-oriented-programming>
2. Крос-платформне програмування (Електронний навчальний курс). [Електронний ресурс] – Режим доступу: <https://elearn.nubip.edu.ua/course/view.php?id=318>