

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ХУДОЖНЬО-ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
Відділення механіко-технологічне

	СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ "Крос-платформне програмування" Освітньо-професійної програми: <u>«Обслуговування програмних систем і комплексів»</u> (назва освітньо-професійної програми) Спеціальність <u>F3 «Комп'ютерні науки»</u> (шифр та назва спеціальності) Галузь знань: <u>F «Інформаційні технології»</u> (шифр та назва галузі знань) Освітньо-професійний ступінь: фаховий молодший бакалавр
Циклова комісія	Комп'ютерних та землевпорядних дисциплін
Курс, семестр	IV курс, VII, VIII семестр
Обсяг	6 кредитів ЄКТС – 180 годин. Аудиторні заняття: лекційні – 40 год., практичні - 50 год., самостійне опрацювання студентів – 90 год.
Мова викладання	українська
Вимоги до початку вивчення	Необхідні знання та навички з дисциплін «Алгоритми та структури даних» «Бази даних»
Відповідність компетентностей та програмних результатів навчання	Загальні компетентності (ЗК): ЗК2. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя. ЗК3. Здатність до абстрактного мислення, аналізу та синтезу. ЗК4. Здатність застосовувати знання у практичних ситуаціях. ЗК5. Знання та розуміння предметної області та розуміння професійної діяльності. ЗК6. Здатність спілкуватися державною мовою як усно, так і письмово. ЗК7. Здатність спілкуватися іноземною мовою. ЗК9. Здатність до пошуку, оброблення та аналізу інформації з різних джерел. Результати навчання (РН): РН05. Розуміти основні методи і технології об'єктно-орієнтованого та компонентного програмування. РН08. Розробляти застосунки, використовуючи сучасні веб-технології. РН10. Знати методології, методи, моделі, процеси і технології життєвого циклу розробки та тестування програмного забезпечення. РН11. Застосовувати сучасні мови програмування та технології для розробки програмного забезпечення розподілених систем. РН15. Розробляти супровідну документацію на різних етапах процесу життєвого циклу розробки програмного забезпечення. Спеціальні компетентності (СК) СК1. Здатність використовувати основні поняття, ідеї та методи фундаментальних наук під час розв'язання складних спеціалізованих задач з комп'ютерних наук в галузі інформаційних технологій. СК2. Здатність використовувати теоретичні та фундаментальні знання в галузі комп'ютерних наук та інформаційних технологій для вирішення різноманітних проблем. СК4. Здатність здійснювати проектування та розробку програмного забезпечення.

	СК5. Здатність застосовувати принципи і методи побудови та використання мережевих технологій. СК6. Здатність застосовувати методи та засоби захисту програмного забезпечення та даних від несанкціонованого доступу в умовах супроводження та експлуатації програмних систем і комплексів. СК7. Здатність проєктувати, розробляти та обслуговувати веб-застосунки з динамічним контентом, використовуючи веб-технології, технології комп'ютерної графіки та анімації. СК8. Здатність застосовувати сучасні методи, технології та інструментальні засоби проєктування й створення програмних систем та їх супроводження. СК10. Здатність адмініструвати системне та прикладне програмне забезпечення під час реалізації процесів життєвого циклу інформаційних систем. СК11. Здатність застосовувати методи та техніки тестування програмного забезпечення впродовж життєвого циклу розробки програмних систем.
Місце дисципліни в освітньо-професійній програмі	Дисципліна «Крос-платформне програмування» належить до циклу професійної та практичної підготовки освітньо-професійної програми спеціальності ФЗ «Комп'ютерні науки». Вона забезпечує формування базових теоретичних знань та практичних навичок, необхідних для розробки програмних застосунків, що функціонують на різних операційних системах і платформах із використанням єдиного програмного коду або спільної логіки. Зміст навчального матеріалу дисципліни спрямований на розвиток у студентів логічного й аналітичного мислення, умінь проєктувати архітектуру крос-платформних застосунків, працювати з інтерфейсами користувача, взаємодіяти з апаратними та програмними можливостями різних платформ, а також створювати ефективні та масштабовані програмні рішення. У процесі вивчення дисципліни студенти набувають практичних навичок використання сучасних крос-платформних технологій і фреймворків, організації спільного коду, забезпечення продуктивності, зручності користування та сумісності застосунків на різних пристроях. Вивчення курсу «Крос-платформне програмування» є важливим етапом підготовки фахівця з комп'ютерних наук, оскільки формує ключові компетенції для розробки сучасних програмних продуктів, орієнтованих на різні операційні системи, мобільні та настільні середовища, і відповідає актуальним вимогам ІТ-індустрії.
Що буде вивчатися	У рамках дисципліни «Крос-платформне програмування» студенти вивчатимуть: Основи крос-платформної розробки програмного забезпечення, включаючи підходи до створення застосунків для різних операційних систем із використанням спільної кодової бази, принципи абстракції платформозалежної логіки, моделі побудови інтерфейсу користувача та архітектурні шаблони крос-платформних застосунків. Окрему увагу буде приділено сучасним крос-платформним фреймворкам і середовищам розробки, організації спільного коду, роботі з компонентами інтерфейсу, обробці подій, доступу до апаратних ресурсів та інтеграції з зовнішніми сервісами. Студенти ознайомляться з принципами побудови настільних і мобільних крос-платформних застосунків, навчатися створювати графічні інтерфейси, працювати з навігацією, анімацією, локалізацією та адаптацією застосунків до різних розмірів екранів і платформ. Курс охоплює питання взаємодії крос-платформних застосунків із серверними частинами, використання REST-API, обміну даними у форматах JSON та XML, а також принципи підключення до баз даних і хмарних сервісів.

	Також розглядатимуться питання продуктивності, безпеки крос-платформних застосунків, управління станом, тестування, налагодження та підготовки застосунків до розгортання на різних платформах. Розділ 1. Основи крос-платформного програмування та архітектури застосунків 1 Поняття крос-платформного програмування 2 Переваги та обмеження крос-платформного підходу 3 Типи крос-платформних застосунків (мобільні, настільні, веб-гібридні) 4 Архітектурні шаблони (MVC, MVVM, MVP) 5 Принципи організації спільного коду 6 Основи об'єктно-орієнтованого програмування у крос-платформному середовищі 7 Робота з колекціями та структурами даних 8 Асинхронне програмування та робота з потоками 9 Подієва модель застосунків 10 Основи створення графічних інтерфейсів користувача Розділ 2. Крос-платформні фреймворки та середовища розробки 1 Огляд сучасних крос-платформних технологій і фреймворків 2 Структура крос-платформного проєкту 3 Робота з компонентами інтерфейсу 4 Макети, стилі та теми оформлення 5 Адаптивні інтерфейси та підтримка різних розмірів екранів 6 Обробка подій користувача 7 Навігація між екранами 8 Робота з ресурсами (зображення, шрифти, локалізація) 9 Налаштування та тестування крос-платформних застосунків Розділ 3. Розробка мобільних і настільних крос-платформних застосунків 1 Життєвий цикл крос-платформного застосунку 2 Створення мобільних застосунків для Android та iOS 3 Створення настільних застосунків для Windows, Linux, macOS 4 Робота з елементами керування 5 Анімації та візуальні ефекти 6 Використання сенсорного вводу та жестів 7 Доступ до файлової системи та локального сховища 8 Інтеграція з апаратними можливостями пристрою Розділ 4. Взаємодія з сервером, дані та безпека 1 Підключення крос-платформних застосунків до серверів 2 Обмін даними через HTTP/HTTPS 3 Робота з REST-сервісами 4 Формати даних JSON та XML 5 Робота з мережевими бібліотеками 6 Основи роботи з базами даних у крос-платформних застосунках 7 Зберігання та кешування даних 8 Основи безпеки крос-платформних застосунків 9 Автентифікація, авторизація, робота з токенами 10 Підготовка застосунків до публікації та розгортання
Чому це цікаво/треба вивчати	Дисципліна «Крос-платформне програмування» є цікавою, оскільки дозволяє зрозуміти, як створюються сучасні програмні застосунки, що працюють одночасно на різних платформах і пристроях — смартфонах, планшетах, персональних комп'ютерах та інших цифрових середовищах. Студенти дізнаються, як за допомогою єдиної кодової бази розробляти програми для кількох операційних систем, зберігаючи продуктивність, зручність користування та сучасний дизайн.

	У процесі вивчення дисципліни студенти відкривають для себе принципи побудови крос-платформної архітектури, створення інтерфейсів користувача, адаптації застосунків до різних екранів і платформ, а також організації взаємодії між клієнтською частиною та серверними сервісами. Курс дає можливість побачити, як функціонують мобільні та настільні застосунки, які ми використовуємо щодня, і як вони інтегруються з мережевими сервісами, API та хмарними платформами. Цікавість дисципліни також полягає у широких можливостях практичного застосування отриманих знань: від створення власних мобільних і настільних програм до розробки комерційних крос-платформних продуктів для бізнесу та користувачів. Студенти працюють із сучасними крос-платформними фреймворками та інструментами розробки, спостерігаючи, як теоретичні концепції перетворюються на практичні навички, що є затребуваними в сучасній IT-індустрії.
Чому можна навчатися	Знання з дисципліни «Крос-платформне програмування» можна буде використовувати для розробки, тестування, розгортання та підтримки програмних застосунків, що функціонують на різних операційних системах і пристроях із використанням спільної кодової бази. Студенти зможуть створювати мобільні та настільні застосунки, проектувати крос-платформні інтерфейси користувача, працювати з архітектурою застосунків, обробляти події та реалізовувати взаємодію з серверними сервісами. У сфері програмної інженерії здобуті навички дозволять створювати сучасні крос-платформні програмні продукти для бізнесу та користувачів, зокрема мобільні застосунки, настільні програми, гібридні рішення та клієнтські частини веб-систем. Студенти навчатися застосовувати сучасні крос-платформні фреймворки та інструменти розробки, інтегрувати застосунки з REST-API та базами даних, забезпечувати зручність користування, продуктивність і безпеку програмних рішень відповідно до вимог сучасної IT-галузі.
Як користуватися набутими знаннями/уміннями	Дисципліна «Крос-платформне програмування» охоплює фундаментальні принципи проектування, розробки та функціонування програмних застосунків, призначених для роботи на різних операційних системах і пристроях із використанням спільної кодової бази. У рамках курсу студенти вивчатимуть архітектуру крос-платформних рішень, підходи до організації програмної логіки та інтерфейсів користувача, механізми адаптації застосунків до різних платформ, а також основи взаємодії з операційною системою та зовнішніми сервісами. Окрему увагу буде приділено принципам створення графічних інтерфейсів, управлінню станом застосунків, роботі з мережевими API та інтеграції крос-платформних застосунків із серверними технологіями. Ця дисципліна є важливою складовою підготовки фахівців у сфері інформаційних технологій, оскільки поєднує знання з програмної інженерії, розробки інтерфейсів користувача та сучасних підходів до створення універсальних програмних продуктів. Студенти здобудуть навички розробки мобільних і настільних крос-платформних застосунків, інтеграції з базами даних і веб-сервісами, використання сучасних фреймворків та інструментів розробки, а також застосування практик, що забезпечують продуктивність, надійність і масштабованість програмних рішень. Курс також дозволить навчитися створювати застосунки для різних платформ, підтримувати їх життєвий цикл та ефективно працювати в умовах сучасного багатоплатформного середовища.
Заняття	Лекційні заняття: формування теоретичних знань щодо принципів крос-платформного програмування, архітектури крос-платформних застосунків, моделей організації спільного коду, побудови інтерфейсів користувача та взаємодії з операційними системами різних платформ. Використання мультимедійних матеріалів, демонстрація роботи крос-платформних застосунків, обговорення сучасних фреймворків, інструментів розробки та актуальних підходів у галузі багатоплатформних програмних рішень. Практичні заняття: закріплення теоретичного матеріалу через розробку крос-платформних застосунків, створення спільної бізнес-логіки, проектування графічних інтерфейсів, роботу з компонентами керування, навігацією, подіями та ресурсами. Студенти виконують завдання з інтеграції застосунків із REST API,

	обробки даних, роботи з локальним сховищем, а також тестування і налагодження застосунків на різних платформах. Лабораторні заняття: застосування отриманих знань під час створення повноцінних крос-платформних програмних продуктів: розробка мобільних і настільних застосунків із єдиною кодовою базою, реалізація архітектурних шаблонів, організація взаємодії між компонентами, інтеграція з серверними сервісами та базами даних, реалізація адаптивних інтерфейсів і підготовка застосунків до розгортання. Інтерактивні заняття: використання навчальних платформ (Google Classroom, Kahoot, Quizizz) для проведення інтерактивних тестів, опитувань і перевірки теоретичних знань, а також платформ для спільної розробки (GitHub Classroom, Replit) з метою колективної роботи над крос-платформними проєктами, аналізу коду та командної взаємодії.
Інформаційне забезпечення	Рекомендовані джерела інформації Andrew Troelsen, Philip Japikse. Pro C# and .NET. — Apress, останнє видання. David Ortinau, James Montemagno. Enterprise Application Patterns using .NET MAUI. — Microsoft Press, 2023. Mark Richards, Neal Ford. Fundamentals of Software Architecture. — O'Reilly, 2020. Adam Freeman. Essential .NET MAUI. — Apress, 2023. Eric Windmill. Flutter in Action / альтернативно React Native in Action (на вибір як приклад крос-платформних підходів). Martin Fowler. Patterns of Enterprise Application Architecture. — Addison-Wesley, 2011. Robert Nystrom. Game Programming Patterns (для розуміння архітектурних підходів, шаблонів і логіки застосунків). Офіційна документація (онлайн): Microsoft Learn / Microsoft Docs — .NET MAUI, .NET, C#, XAML. Flutter Documentation / React Native Documentation — огляд альтернативних крос-платформних фреймворків. Android Developers Documentation — життєвий цикл, UI, робота з ресурсами. Apple Developer Documentation — загальні принципи UI та життєвого циклу застосунків. Допоміжна література Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship. — Prentice Hall, 2008. Robert C. Martin. Clean Architecture. — Prentice Hall, 2017. Joshua Bloch. Effective Java (для загальних принципів якості коду та архітектури). Tom Hombergs. Get Your Hands Dirty on Clean Architecture. — 2019. Kyle Loudon. Mastering Algorithms with C / Java / Python — для алгоритмічного мислення та структур даних. Для взаємодії з сервером і даними: RESTful Web APIs — Leonard Richardson, Mike Amundsen, Sam Ruby. — O'Reilly. Документація SQLite / MySQL / PostgreSQL. Для командної розробки: GitHub Guides — інструменти контролю версій і командної роботи. Git Documentation — Git Flow, pull requests, code review.

Поточний контроль	Поточний контроль здійснюється викладачем у формі усного спілкування зі студентами, письмового, тестового експрес-контролю (усні відповіді, розв'язання завдань студентами за ПК та на місцях, самостійні роботи, доповіді, презентації виконаних завдань, досліджень, тести тощо) і має за мету перевірку ступеня засвоєння певного навчального матеріалу, а також рівня оволодіння вміннями та навичками.
Семестровий контроль	Підсумковий контроль здійснюється у формі екзамену

