

**ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО**

Навчально-науковий інститут інформаційних та освітніх технологій

Кафедра прикладної математики та інформатики

ЗАТВЕРДЖЕНО:

на засіданні кафедри
прикладної математики та інформатики
(протокол № 8 від «23» лютого 2025 р.)

Ф-КАТАЛОГ

ВИБІРКОВИХ НАВЧАЛЬНИХ ДИСЦИПЛІН

ЦИКЛУ ПРОФЕСІЙНОЇ ПІДГОТОВКИ

для здобувачів ступеня бакалавр

**за освітньо-професійною програмою
«Прикладна математика»**

за спеціальністю 113 «Прикладна математика»

ЗМІСТ

ПЕРЕДМОВА	3
ПЕРЕЛІК ОСВІТНІХ КОМПОНЕНТІВ	4
Принципи та інструменти веб-дизайну.....	4
Програмні засоби мультимедіа	6
Методи статистичного виведення	8
Створення та розвиток ІТ-продуктів	10
Розробка серверної частини веб-додатків на основі FastAPI.....	12
Основи криптології	14
Проектування інформаційних систем.....	16
Фреймворки для розробки клієнтської частини веб-додатків	18
Комп'ютерні мережі.....	20
Теорія автоматичного керування.....	22
Технічні засоби автоматизації.....	24
Підготовка документів у системі комп'ютерної верстки LaTeX	26
Основи візуалізації даних	28
Функціональне програмування (мова програмування Haskell)	30
Програмування мікропроцесорної техніки	32
Мова програмування Assembler для мікропроцесорів x86	34
Тестування програмного забезпечення.....	36
Обчислювальна геометрія та комп'ютерна графіка.....	38

ПЕРЕДМОВА

Відповідно до пункту 15 частини першої статті 62 Закону України «Про вищу освіту» (№ 1556-VII від 01.07.2014), вибіркові дисципліни – це дисципліни вільного вибору студентів для певного рівня вищої освіти, спрямовані на забезпечення загальних та спеціальних (фахових) компетентностей відповідної освітньої програми. Обсяг вибірових навчальних дисциплін повинен становити не менше 25% від загальної кількості кредитів ЄКТС, передбачених для даного рівня освіти.

Вибіркові фахові навчальні дисципліни з Ф-Каталогу здобувачі вищої освіти обирають відповідно до «Положення про порядок формування варіативної складової індивідуального навчального плану студента Черкаського національного університету імені Богдана Хмельницького» (<https://drive.google.com/file/d/1slAN3uuIOx3swmySQSCrrdptloiaB9oh/view>).

Каталог містить анотований перелік освітніх компонентів, які пропонуються для обрання здобувачами 1-3 курсів першого (бакалаврського) рівня вищої освіти згідно навчального плану на наступний навчальний рік (на 2-4 курси відповідно).

Здійснення вибору студентами навчальних дисциплін зі сформованого Ф-Каталогу організовується випусковою кафедрою відповідно до графіку.

ПЕРЕЛІК ОСВІТНІХ КОМПОНЕНТІВ

Назва дисципліни	Принципи та інструменти веб-дизайну
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Для вивчення дисципліни потрібно знати основи HTML, CSS, основи роботи з графічними редакторами, базові принципи дизайну та адаптивності сайтів, базові принципи роботи з текстовими редакторами та графічними файлами, а також ознайомитися з поняттями інтерфейсу користувача і візуального дизайну.
Предмет навчання (Що буде вивчатися)	– основи дизайну інтерфейсу користувача (UI) та досвіду користувача (UX); – проведення UX-досліджень; – інструменти для створення макетів веб-сторінок (Figma); – створення прототипів та анімованих макетів; – проведення аналізу цільової аудиторії; – типографіка та кольорові схеми у веб-дизайні; – принципи адаптивного дизайну для різних пристроїв; – робота з графічними елементами та ілюстраціями; – веб-доступність та інклюзивний дизайн; – оптимізація веб-дизайну для швидкого завантаження; – тренди сучасного веб-дизайну; – оформлення робіт у кейс та створення портфолію.
Мета (Чому це цікаво/потрібно вивчати)	Вивчення принципів та інструментів веб-дизайну дозволяє створювати привабливі, функціональні та доступні веб-сайти, що відповідають потребам користувачів. Це важливо для веб-додатків та для створення ефективних цифрових продуктів, які привертають увагу, забезпечують комфортне використання і підвищують взаємодію з аудиторією
Програмні результати (Чому можна навчитися)	Можна навчитися створювати інтуїтивно зрозумілі та естетично привабливі веб-інтерфейси, працювати з інструментами веб-дизайну, застосовувати принципи UX/UI, розуміти основи адаптивного та інклюзивного дизайну, проводити аналіз цільової аудиторії, працювати з типографікою та графічними елементами, а також інтегрувати графіку та мультимедіа у веб-проекти для покращення користувацького досвіду

Компетентності <i>(Як можна користуватися набутими знаннями і вміннями)</i>	Набуті знання і вміння можна використовувати для створення веб-сайтів та веб-додатків, що відповідають сучасним стандартам дизайну та зручності використання. Це відкриває можливості для роботи у сфері веб-дизайну, фріланс-проектів або стартапів. Знання принципів UX/UI дозволять ефективно взаємодіяти з клієнтами та командою розробників, забезпечуючи реалізацію їхніх вимог. Також ці навички можуть бути корисними для просування особистих проектів у цифровому середовищі, поліпшуючи їхню видимість та залучаючи більше користувачів
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та матеріали практичних завдань, навчальні посібники
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Програмні засоби мультимедіа
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Базові знання з програмування, основи роботи з графікою, аудіо та відео форматами, знання алгоритмів обробки зображень та звуку
Предмет навчання (Що буде вивчатися)	– основи мультимедійних технологій; – програмні засоби для обробки графіки, аудіо та відео; – формати файлів для мультимедіа; – алгоритми стиснення та кодування даних; – інтеграція мультимедійних даних у програми та веб-додатки; – обробка та редагування мультимедійних файлів; – використання спеціалізованих бібліотек для роботи з мультимедіа; – створення інтерактивних мультимедійних додатків; – візуалізація даних та анімація.
Мета (Чому це цікаво/потрібно вивчати)	Мультимедійні технології використовуються у багатьох сферах – від веб-дизайну до кіно та відеоігор; їх вивчення необхідне для створення якісного контенту та інтерактивних додатків
Програмні результати (Чому можна навчитися)	– обробляти та редагувати мультимедійні файли; – використовувати сучасні інструменти для створення мультимедіа; – інтегрувати мультимедійні компоненти у програмні продукти; – створювати інтерактивний та візуально привабливий контент; – оптимізувати мультимедійні файли для різних платформ.
Компетентності (Як можна користуватися набутими знаннями і уміннями)	Знання дозволять створювати мультимедійні додатки; інтегрувати графіку та звук у проекти; працювати в індустріях, що пов'язані з розробкою ігор, кіно, рекламних продуктів та освітнього контенту
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та матеріали практичних завдань, навчальні посібники

Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Методи статистичного виведення
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Базові знання теорії ймовірностей; основи математичної статистики; знання основ алгебри та математичного аналізу
Предмет навчання (Що буде вивчатися)	– основи статистичного аналізу; – методи оцінювання параметрів; – перевірка статистичних гіпотез; – методи побудови довірчих інтервалів; – методи лінійної та нелінійної регресії; – дисперсійний аналіз; – Байєсове виведення; – методи вибірових досліджень; – статистичне моделювання та симуляції.
Мета (Чому це цікаво/потрібно вивчати)	Статистичні методи є основою для аналізу даних, прийняття рішень та прогнозування, що робить їх важливими у різних галузях, від економіки до медицини та технологій
Програмні результати (Чому можна навчитися)	– аналізувати дані за допомогою статистичних методів; – будувати моделі для прогнозування та інтерпретації результатів; – перевіряти гіпотези та робити висновки на основі даних; – застосовувати різні методи оцінювання та регресійного аналізу; – працювати з реальними даними та статистичними програмними засобами.
Компетентності (Як можна користуватися набутими знаннями і уміннями)	Застосовувати статистичні методи для аналізу даних у наукових дослідженнях, маркетингу, фінансах, медицини та будь-яких сферах, де важлива аналітика та прогнозування; створювати прогнози, інтерпретувати дані та приймати обґрунтовані рішення на їх основі
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та матеріали практичних завдань, навчальні посібники
Матеріально-технічне	Аудиторія теоретичного навчання, комп'ютерний клас для

забезпечення	виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Створення та розвиток ІТ-продуктів
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	6 кредитів ЄКТС: – лекції – 28 годин, – лабораторні заняття – 32 години, – самостійна робота – 120 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Розуміння життєвого циклу програмного забезпечення, основи проектного менеджменту, а також знання принципів роботи ринку ІТ-продуктів і розуміння бізнес-процесів.
Предмет навчання (Що буде вивчатися)	Дисципліна розрахована на ознайомлення з базовими компетенціями trainee продактменеджера: – організація циклу розвитку продукту; – організація процесів у продуктовій команді; – основи продуктового дизайну; – основи роботи з інструментами аналізу ринку та цільової аудиторії; – основи маркетингу в продуктовому ІТ; – основи продуктової аналітики; – основи роботи з інструментами та практиками прийняття рішень в продуктових командах.
Мета (Чому це цікаво/потрібно вивчати)	Створення ІТ-продуктів є ключовим навичкою в сучасній економіці, оскільки попит на інноваційні рішення в ІТ постійно зростає. Вивчення цієї дисципліни дає можливість зрозуміти повний цикл розробки продукту – від ідеї до його виходу на ринок, що особливо важливо для тих, хто прагне стати успішним підприємцем або продукт-менеджером.
Програмні результати (Чому можна навчитися)	Студент знає: – які етапи створення та розвитку ІТ-продуктів існують, а також критерії, за якими визначається їхня успішність; – з яких фахівців можуть складатися продуктові команди ІТ-стартапів, їхні ролі в команді; – хто такий продактменеджер, його основні функції під час розробки продукту, аспекти взаємодії та менеджменту в продуктових компаніях; – що таке мінімально життєздатний продукт, які існують способи валідації ідей та як працювати з тестуванням гіпотез; – базові поняття маркетингу в сфері продуктового ІТ, способи дослідження та впливу на цільову аудиторію продукту; – особливості роботи технічної та нетехнічної складової

	продуктової команди, аспекти застосування мов програмування в залежності від продукту; – основну термінологію сфери продуктового ІТ. Студент вміє: – визначати успішність ІТ-продуктів за допомогою спеціальних метрик; – формувати продуктову команду для запуску власного стартапу; – валідувати свої ідеї за допомогою кількісних та якісних опитувань, а також завдяки тестуванню гіпотез; – оперувати основними маркетинговими інструментами для просування продукту; – взаємодіяти з технічними членами команди, розуміючи, яку роль та функцію вони відіграють в розробці продукту; – мислити продуктовими та підприємницькими категоріями; – виконувати базові компетенції на рівні trainee продактменеджера.
Компетентності <i>(Як можна користуватися набутими знаннями і уміннями)</i>	Навчальний модуль розрахований на ознайомлення з базовими компетенціями trainee продактменеджера: – організація циклу розвитку продукту; – організація процесів у продуктивній команді; – основи продуктового дизайну; – основи роботи з інструментами аналізу ринку та цільової аудиторії; – основи маркетингу в продуктовому ІТ; – основи продуктової аналітики; – основи роботи з інструментами та практиками прийняття рішень в продуктових командах.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та матеріали практичних завдань, навчальні посібники
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Розробка серверної частини веб-додатків на основі FastAPI
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	6 кредитів ЄКТС: – лекції – 28 годин, – лабораторні заняття – 32 години, – самостійна робота – 120 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Володіння мовою програмування Python; базові знання про HTTP-протокол, REST API та принципи клієнт-серверної взаємодії; розуміння основ роботи з базами даних.
Предмет навчання (Що буде вивчатися)	– основи FastAPI та асинхронного програмування в Python; – створення RESTful API endpoints з використанням різних HTTP-методів; – валідація даних за допомогою Pydantic моделей; – робота з базами даних через ORM (SQLAlchemy); – аутентифікація та авторизація користувачів (JWT токени, OAuth2); – документування API за допомогою OpenAPI та Swagger; – обробка помилок та винятків; – тестування серверних додатків (pytest, TestClient).
Мета (Чому це цікаво/потрібно вивчати)	FastAPI є одним з найсучасніших та найшвидших Python фреймворків для розробки веб-API, що активно використовується у стартапах та великих технологічних компаніях. Вивчення цієї дисципліни дає можливість створювати серверні додатки з автоматичною генерацією документації, вбудованою валідацією даних та підтримкою асинхронності. Знання FastAPI дозволяє швидко прототипувати та масштабувати веб-сервіси, а також є цінною навичкою для розробки мікросервісної архітектури та API-first додатків.
Програмні результати (Чому можна навчитися)	– створювати повноцінні RESTful API з використанням FastAPI; – розробляти асинхронні обробники запитів для підвищення продуктивності; проектувати та валідувати моделі даних за допомогою Pydantic; – інтегрувати додаток з реляційними базами даних (PostgreSQL, MySQL); – імплементувати систему аутентифікації з JWT токенами; – обробляти файли, форми та JSON дані; – писати автоматичні тести для API endpoints; – документувати API з використанням вбудованих інструментів.

Компетентності (Як можна користуватися набутими знаннями і вміннями)	Розробка високопродуктивних RESTful API; асинхронне програмування для обробки великої кількості запитів; проектування архітектури серверних додатків; інтеграція з базами даних та зовнішніми сервісами; забезпечення безпеки веб-API; розробка документації та API специфікацій.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та методичні матеріали лабораторних робіт; офіційна документація FastAPI; підручники з Python веб-розробки; приклади коду та шаблони проєктів; навчальні відеоматеріали; онлайн-ресурси та туторіали; Case Studies реальних проєктів на FastAPI.
Матеріально-технічне забезпечення	Комп'ютерний клас з встановленим програмним забезпеченням (Python 3.9+, FastAPI, SQLAlchemy, PostgreSQL/MySQL, IDE: PyCharm/VS Code), аудиторія для проведення лекцій з мультимедійним обладнанням, проєктор, навчальна та технічна література, презентаційні матеріали.
Вид семестрового контролю	Залік

Назва дисципліни	Основи криптології
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Знання основ теорії чисел, алгебри, дискретної математики, теорії ймовірностей та основ алгоритмів
Предмет навчання (Що буде вивчатися)	– основи симетричних та асиметричних криптосистем; – шифрування та дешифрування даних; – криптографічні хеш-функції; – цифрові підписи; – криптографічні протоколи; – атаки на криптосистеми та методи захисту; – квантова криптографія.
Мета (Чому це цікаво/потрібно вивчати)	Криптологія є важливою частиною сучасних інформаційних технологій, оскільки забезпечує безпеку та конфіденційність даних, які відіграють ключову роль у всіх сферах життя – від особистої комунікації до глобальної економіки
Програмні результати (Чому можна навчитися)	– розуміти принципи сучасних криптосистем; – використовувати криптографічні методи для захисту інформації; – аналізувати безпеку систем та виявляти потенційні загрози; – застосовувати криптографію в реальних проєктах та програмах; – розробляти безпечні протоколи передачі даних.
Компетентності (Як можна користуватися набутими знаннями і вміннями)	Розробка та забезпечення безпеки інформаційних систем, робота у сфері кібербезпеки, розробка програмного забезпечення для захисту даних, консалтинг у питаннях інформаційної безпеки, аналітика та аудит систем безпеки
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та матеріали практичних завдань, навчальні посібники
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали

Вид семестрового контролю	Залік

Назва дисципліни	Проектування інформаційних систем
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Знання основ програмування, баз даних, алгоритмів та структур даних. Розуміння життєвого циклу розробки програмного забезпечення, основ об'єктно-орієнтованого програмування та принципів роботи клієнт-серверних додатків.
Предмет навчання (Що буде вивчатися)	Дисципліна спрямована на формування комплексних знань та практичних навичок у проектуванні сучасних інформаційних систем: – методології та підходи до проектування інформаційних систем; – аналіз вимог та специфікація функціональних і нефункціональних характеристик; – архітектурні патерни та стилі побудови інформаційних систем; – проектування баз даних та моделювання предметної області; – UML-моделювання та використання CASE-засобів; – проектування інтерфейсів користувача та UX/UI принципи; – інтеграція систем та API-проектування; – безпека інформаційних систем та управління доступом; – тестування та документування проєктних рішень.
Мета (Чому це цікаво/потрібно вивчати)	Проектування інформаційних систем є фундаментальною компетенцією для сучасних ІТ-фахівців, оскільки якість проєктних рішень визначає успішність, масштабованість та надійність програмних продуктів. Вивчення дисципліни дозволяє опанувати системний підхід до створення складних інформаційних систем, навчитися приймати обґрунтовані архітектурні рішення та передбачати довгострокові наслідки проєктних рішень. Це критично важливо для побудови кар'єри як системного аналітика, архітектора програмного забезпечення або провідного розробника.
Програмні результати (Чому можна навчитися)	– проводити аналіз предметної області та формалізувати вимоги до інформаційної системи; – обирати оптимальну архітектуру системи відповідно до

	поставлених задач та обмежень; – створювати концептуальні, логічні та фізичні моделі даних; – використовувати UML-діаграми для документування проєктних рішень (діаграми класів, послідовності, компонентів, розгортання тощо); – застосовувати патерни проєктування для вирішення типових проблем; – презентувати та захищати архітектурні рішення перед стейкхолдерами проєкту.
Компетентності <i>(Як можна користуватися набутими знаннями і вміннями)</i>	Дисципліна формує ключові компетентності для проєктування інформаційних систем: системний аналіз та формалізація бізнес-вимог; архітектурне проєктування; моделювання даних та проєктування баз даних; забезпечення якості, безпеки та продуктивності систем; документування та комунікація проєктних рішень.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та методичні матеріали лабораторних робіт; підручники з проєктування інформаційних систем.
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Фреймворки для розробки клієнтської частини веб-додатків
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики.
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	володіння основами веб-технологій (HTML5, CSS3, JavaScript). Розуміння принципів об'єктно-орієнтованого. Знання основ роботи з DOM. Базові навички роботи з інструментами розробки (npm/yarn, браузерні DevTools). Розуміння принципів REST API та взаємодії клієнта з сервером.
Предмет навчання (Що буде вивчатися)	– порівняльний аналіз популярних фреймворків (React, Vue.js, Angular); – компонентний підхід до розробки інтерфейсів; управління станом додатка (Redux, Vuex, Pinia, NgRx); – робота з віртуальним DOM та оптимізація рендерингу; – маршрутизація в односторінкових додатках (SPA); – робота з формами та валідація даних на клієнті; – інтеграція з REST API та GraphQL; – TypeScript для типобезпечної розробки; стилізація компонентів (CSS Modules, Styled Components, Tailwind CSS); – тестування frontend додатків (Jest, React Testing Library, Cypress); – збірка та оптимізація проєктів (Webpack, Vite); – сучасні паттерни та best practices frontend розробки.
Мета (Чому це цікаво/потрібно вивчати)	Сучасні JavaScript фреймворки є фундаментом для створення складних інтерактивних веб-додатків і становлять основу frontend розробки в ІТ-індустрії. Вивчення цієї дисципліни дозволяє опанувати інструменти, які використовуються в компаніях по всьому світу для розробки веб-застосунків. Знання популярних фреймворків відкриває широкі можливості для працевлаштування як frontend або full-stack розробника, дозволяє створювати сучасні користувацькі інтерфейси з високою продуктивністю та підтримуваністю коду. Це критично важлива навичка для тих, хто прагне працювати над створенням веб-продуктів, мобільних додатків або десктопних застосунків на базі веб-технологій.
Програмні результати (Чому можна навчитися)	– створювати повноцінні односторінкові додатки з використанням сучасних фреймворків;

	– розробляти перевикористовувані та підтримувані компоненти; – імплементувати складну логіку управління станом додатка; – налаштовувати маршрутизацію та навігацію в SPA; – інтегрувати frontend з REST API та GraphQL endpoints; – створювати responsive та adaptive інтерфейси; – писати unit та integration тести для компонентів; – працювати з формами, валідацією та обробкою помилок; – реалізовувати аутентифікацію та авторизацію на клієнті.
Компетентності <i>(Як можна користуватися набутими знаннями і вміннями)</i>	Розробка складних користувацьких інтерфейсів; архітектурне проектування frontend додатків; управління станом та даними у проєктах; оптимізація продуктивності веб-застосунків; забезпечення якості через автоматизоване тестування; робота в команді з використанням сучасних workflow; інтеграція з backend сервісами та API.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та методичні матеріали лабораторних робіт; офіційна документація React, Vue.js; підручники з JavaScript та TypeScript; онлайн-курси та відеоматеріали; приклади коду та starter templates; навчальні проєкти з відкритим вихідним кодом; статті та блоги провідних frontend розробників; Case Studies реальних додатків на різних фреймворках.
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали.
Вид семестрового контролю	Залік

Назва дисципліни	Комп'ютерні мережі
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики.
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Знання основ інформатики та архітектури комп'ютерних систем. Розуміння принципів роботи операційних систем та командного рядка. Базові знання математики (системи числення, логічні операції, основи теорії ймовірностей). Навички роботи з технічною документацією та англomовними термінами. Розуміння основних концепцій програмування та алгоритмізації.
Предмет навчання (Що буде вивчатися)	– еталонна модель OSI та стек протоколів TCP/IP; – фізичний рівень передачі даних та середовища передачі; – канальний рівень та технології локальних мереж (Ethernet, Wi-Fi); – мережевий рівень та протокол IP (IPv4, IPv6); – маршрутизація та протоколи маршрутизації (RIP, OSPF, BGP); – транспортний рівень та протоколи TCP/UDP; – прикладний рівень та мережеві сервіси (DNS, DHCP, HTTP, FTP); – основи мережевої безпеки та криптографії; – технології віртуальних приватних мереж (VPN); – якість обслуговування (QoS) та управління трафіком; – бездротові та мобільні мережі; – адміністрування та моніторинг мережевої інфраструктури; – хмарні технології та SDN (Software-Defined Networking); – діагностика та усунення неполадок у мережах.
Мета (Чому це цікаво/потрібно вивчати)	Комп'ютерні мережі є основою сучасної цифрової інфраструктури, забезпечуючи комунікацію між мільярдами пристроїв у всьому світі. Розуміння принципів роботи мереж є критично важливим для будь-якого ІТ-фахівця, оскільки практично всі сучасні додатки та сервіси функціонують у мережевому середовищі. Вивчення цієї дисципліни дозволяє зрозуміти, як працює Інтернет, як передаються дані між комп'ютерами, як забезпечується безпека та надійність комунікацій. Ці знання необхідні для кар'єри мережевого інженера, системного адміністратора, DevOps спеціаліста, фахівця з кібербезпеки, а також є фундаментальними для розробників будь-якого профілю.

Програмні результати (Чому можна навчитися)	– аналізувати та проектувати топологію локальних мереж; – виконувати розрахунки IP-адресації та створювати схеми підмереж; – налаштовувати базову конфігурацію мережевого обладнання (маршрутизатори, комутатори); – використовувати утиліти діагностики мережі (ping, traceroute, netstat, nslookup); – аналізувати мережевий трафік за допомогою Wireshark; – налаштовувати статичну та динамічну маршрутизацію; – конфігурувати мережеві сервіси (DNS, DHCP); – забезпечувати базову безпеку мережевої інфраструктури; – налаштовувати VPN з'єднання; – діагностувати та усувати типові мережеві проблеми; – працювати з протоколами прикладного рівня; – моніторити стан мережевого обладнання; – застосовувати best practices мережевого адміністрування; – читати та інтерпретувати RFC специфікації.
Компетентності (Як можна користуватися набутими знаннями і уміннями)	Проектування та побудова локальних та глобальних мереж; адміністрування мережевої інфраструктури; налаштування та підтримка мережевого обладнання; діагностика та усунення мережевих неполадок; забезпечення мережевої безпеки; оптимізація продуктивності мережевих систем; планування розвитку мережевої інфраструктури; інтеграція хмарних та традиційних мережевих рішень; моніторинг та аналіз мережевого трафіку; впровадження політик якості обслуговування; консультування з питань мережевих технологій; технічна підтримка користувачів мережі.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та методичні матеріали лабораторних робіт; підручники з комп'ютерних мереж (Таненбаум, Куроуз-Росс); RFC документація протоколів; документація мережевого обладнання; відеокурси та інтерактивні симулятори; схеми та топології мережевих рішень; Case Studies корпоративних мереж; онлайн-ресурси та технічні блоги.
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Теорія автоматичного керування
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Знання математичного аналізу, диференціальних рівнянь, лінійної алгебри, основ фізики, теорії ймовірностей, а також базові знання з програмування
Предмет навчання (Що буде вивчатися)	– принципи автоматичного керування; – аналіз і проектування систем автоматичного керування; – стійкість та динаміка систем; – методи керування лінійними та нелінійними системами; – передаточні функції та частотні характеристики; – зворотний зв'язок і його роль в системах керування; – оптимізація параметрів систем керування; – адаптивні та інтелектуальні системи керування.
Мета (Чому це цікаво/потрібно вивчати)	Теорія автоматичного керування є ключовою для розуміння та розробки сучасних технологій у промисловості, робототехніці, авіації та багатьох інших галузях, де необхідно забезпечити точне керування процесами та системами
Програмні результати (Чому можна навчитися)	– аналізувати та моделювати системи автоматичного керування; – проектувати стійкі та ефективні системи; – застосовувати сучасні методи керування для складних динамічних систем; – оптимізувати параметри систем для досягнення найкращих результатів; – використовувати спеціалізовані програмні засоби для моделювання та аналізу систем.
Компетентності (Як можна користуватися набутими знаннями і вміннями)	Застосування теорії автоматичного керування в розробці роботів, автономних транспортних засобів, систем керування виробничими процесами, енергетичними установками, а також у будь-яких технологічних процесах, що потребують точного й ефективного керування
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та матеріали практичних завдань, навчальні посібники

Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Технічні засоби автоматизації
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Знання основ електротехніки, фізики, математичного аналізу, а також базові знання з програмування та інформаційних технологій
Предмет навчання (Що буде вивчатися)	– класифікація технічних засобів автоматизації; – принципи роботи автоматизованих систем; – сенсори та виконавчі механізми; – системи управління технологічними процесами; – програмовані логічні контролери (PLC); – інтерфейси людина-машина (HMI); – протоколи зв'язку в автоматизації; – проектування автоматизованих систем.
Мета (Чому це цікаво/потрібно вивчати)	Технічні засоби автоматизації є невід'ємною частиною сучасної промисловості та виробництва, дозволяючи підвищувати продуктивність, знижувати витрати і покращувати якість продукції, а також забезпечуючи безпеку технологічних процесів
Програмні результати (Чому можна навчитися)	– розуміти принципи роботи різних технічних засобів автоматизації; – проектувати та впроваджувати автоматизовані системи; – аналізувати і оптимізувати технологічні процеси; – використовувати сучасні засоби автоматизації для підвищення ефективності виробництва; – інтегрувати різні компоненти автоматизації в єдину систему.
Компетентності (Як можна користуватися набутими знаннями і уміннями)	Застосування знань у розробці, налаштуванні та експлуатації автоматизованих систем в промислових, енергетичних, транспортних та інших галузях, що потребують автоматизації процесів для підвищення ефективності та безпеки
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та матеріали практичних завдань, навчальні посібники

Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Підготовка документів у системі комп'ютерної верстки LaTeX
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Курс базується на поняттях, що вивчаються в дисциплінах: інформатика (робота у операційній системі Windows чи Linux, знайомство з офісними пакетами)
Предмет навчання (Що буде вивчатися)	– огляд систем TeX та LaTeX; – рисунки та таблиці; – оформлення документу в цілому; – робота з бібліографією; – створення та оформлення презентацій та резюме (Curriculum Vitae); – розробка макрокоманд. Оформлення складних математичних формул та лістингів програм; – використання стандартних та розроблених макросів для генерації рисунків та схем.
Мета (Чому це цікаво/потрібно вивчати)	Засвоєння студентами основних теоретичних відомостей та практичних вмінь з курсу. Підготувати студента до ефективного використання системи LaTeX для задач набору тексту і підготовки статей, включаючи набір тексту з використанням кількох мов, нумерацію розділів і формул, перехресні посилання, розміщення ілюстрацій і таблиць на сторінці, ведення бібліографії
Програмні результати (Чому можна навчитися)	– знання та практичні навички з використання системи LaTeX для розробки документів наукового змісту; – знання основних понять та визначень форматування документу, складових документу;
Компетентності (Як можна користуватися набутими знаннями і вміннями)	Вміння розробляти документи з використанням автоматичної генерації змісту, списку ілюстрацій, таблиць тощо; механізму роботи з перехресними посиланнями на формули, таблиці, ілюстрації, їх номер або сторінку; механізм цитування бібліографічних джерел, роботи з бібліографічними картотеками; засобів автоматичного розміщення ілюстрацій; оформлення математичних формул, можливості набирати багаторядкові формули; оформлення хімічних формул і структурних схем молекул органічної і неорганічної хімії; оформлення графів, схем,

	діаграм, синтаксичних графів; оформлення алгоритмів, початкового програмного коду (які можуть включатися в текст безпосередньо зі своїх файлів) з синтаксичним підсвічуванням; розбивки документа на окремі частини (тематичні карти).
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та матеріали практичних завдань, навчальні посібники
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Основи візуалізації даних
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: – лекції – 14 годин, – лабораторні заняття – 26 годин, – самостійна робота – 80 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Курс є адаптивним і може пропонуватись як студентам спеціальностей природничого напрямку, так і студентам спеціальностей технічного напрямку. Базова частина курсу базується на поняттях, що вивчаються в дисциплінах: – інформатика (робота у операційній системі Windows чи Linux, знайомство з офісними пакетами, робота з програмним забезпеченням); – основи вищої математики (поняття точки, функції). Додатково, для студентів, що будуть опрацьовувати додаткові розділи дисципліни, передбачається попереднє знайомство дисциплін: – математична статистика; – мови програмування (досить базових знань з однієї з мов: Python, Java, JavaScript, Go тощо) чи інструментальні засоби математичного моделювання (передбачаються базові знання роботи у одній з систем: MatLab, R, MathCad тощо).
Предмет навчання (Що буде вивчатися)	– основні елементи візуального подання даних; – основні типи графіків та діаграм; – ієрархічні дані; – графи; – географічні дані; – статистична обробка даних; – потокові дані. Моделі; – діаграми Вороного та теплові карти. Довільні фігури..
Мета (Чому це цікаво/потрібно вивчати)	Засвоєння студентами основних теоретичних відомостей та практичних вмінь з курсу. Ознайомити студента з основними елементами візуального представлення даних. Підготувати студента до ефективного використання інструментів для візуалізації даних різної природи з метою створення графічного представлення різних типів: графіків, гістограм, діаграм, графів, карт, схем тощо.
Програмні результати (Чому можна навчитися)	Знання та практичні навички з методів та засобів візуалізації даних. Додатково – знання та практичні навички зі статистичної обробки даних, використання спеціалізованих бібліотек однієї з мов програмування:

	Python, Java, JavaScript, Go тощо. Знання основних понять та елементів графічного відображення даних.
Компетентності <i>(Як можна користуватися набутими знаннями і вміннями)</i>	Вміння використовувати інструменти (додатково – мови програмування та спеціалізовані бібліотеки) для побудови різних типів візуального відображення даних: графіків, гістограм, діаграм, графів, карт, схем тощо.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та матеріали практичних завдань, навчальні посібники
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Функціональне програмування (мова програмування Haskell)
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики.
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	6 кредитів ЄКТС: – лекції – 28 годин, – лабораторні заняття – 32 години, – самостійна робота – 120 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Знання основ програмування та алгоритмізації на будь-якій мові програмування. Розуміння базових структур даних (списки, дерева, масиви) та алгоритмів. Знання основ дискретної математики (логіка, теорія множин, комбінаторика). Базові знання математичного аналізу та лінійної алгебри.
Предмет навчання (Що буде вивчатися)	– основні концепції функціонального програмування та його філософія; – чисті функції та відсутність побічних ефектів; – система типів та виведення типів у Haskell; – базові типи даних та алгебраїчні типи даних; – функції вищого порядку та каррування; – рекурсія та хвостова рекурсія; – pattern matching та guards; – лінійне обчислення та нескінченні структури даних; – функтори, аплікативні функтори та монади; – робота зі списками та List Comprehensions; – модулі та організація коду в Haskell; – введення-виведення та взаємодія з зовнішнім світом; – типові класи та поліморфізм; – парсинг та робота з текстовими даними; – практичне застосування Haskell для розв'язання задач.
Мета (Чому це цікаво/потрібно вивчати)	Функціональне програмування представляє фундаментально інший підхід до розробки програмного забезпечення, що базується на математичних принципах та декларативному стилі мислення. Вивчення Haskell як еталонної функціональної мови дозволяє розвинути абстрактне мислення, глибше зрозуміти природу обчислень та навчитися писати більш надійний, композиційний та легко тестований код. Концепції функціонального програмування активно проникають у сучасні mainstream мови (JavaScript, Python, Java, C#), тому їх розуміння робить програміста більш універсальним. Haskell широко використовується у фінансовій індустрії, блокчейн-технологіях, компіляторах та системах, де критична коректність програм.

Програмні результати (Чому можна навчитися)	– писати програми на мові Haskell у функціональному стилі; – використовувати рекурсію для розв’язання задач; – застосовувати функції вищого порядку (map, filter, fold); – працювати з pattern matching для деконструкції даних; – створювати та використовувати лінійні структури даних; – виконувати введення-виведення у функціональному стилі; – використовувати типові класи для створення поліморфного коду; – оптимізувати функціональний код; – читати та розуміти чужий функціональний код.
Компетентності (Як можна користуватися набутими знаннями і вміннями)	Дисципліна формує унікальні компетенції функціонального програміста: розробка надійного та коректного програмного забезпечення; застосування математичних підходів до програмування; проектування композиційних та модульних систем; створення domain-specific languages (DSL); розробка компіляторів та інтерпретаторів; робота з паралельними та конкурентними обчисленнями; формальна верифікація програм; розробка фінансових та блокчейн систем; аналіз та трансформація коду; використання функціональних патернів у будь-яких мовах програмування.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та методичні матеріали лабораторних робіт; офіційна документація Haskell та GHC; онлайн-курси та відеоматеріали; збірки задач та вправ з функціонального програмування; наукові статті про застосування Haskell; приклади коду з відкритих проєктів; туторіали та блоги Haskell спільноти; документація бібліотек Hackage.
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп’ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Програмування мікропроцесорної техніки
Кафедра, яка забезпечує викладання	Автоматизації та комп'ютерно-інтегрованих технологій
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	6 кредитів ЄКТС: – лекції – 28 годин, – лабораторні заняття – 32 години, – самостійна робота – 120 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Знання основ програмування на мовах високого рівня (C/C++, Python). Розуміння систем числення (двійкова, шістнадцяткова) та логічних операцій. Базові знання з архітектури комп'ютерних систем та принципів роботи процесорів. Розуміння основних структур даних та алгоритмів. Знання основ цифрової електроніки та логічних схем. Вміння читати технічну документацію та схеми електронних пристроїв.
Предмет навчання (Що буде вивчатися)	– архітектура мікропроцесорів та мікроконтролерів; – система команд та асемблер для ARM Cortex-M; – організація пам'яті та адресація у мікроконтролерах; – введення-виведення та робота з периферією; – цифрові порти введення-виведення (GPIO); – таймери, лічильники та ШІМ (PWM); – переривання та їх обробка; – аналого-цифрові та цифро-аналогові перетворювачі (АЦП/ЦАП); – послідовні інтерфейси (UART, SPI, I2C); – програмування мікроконтролерів на C/C++; – робота з датчиками та виконавчими механізмами; – енергозбереження у вбудованих системах; – налагодження та тестування embedded програм; – розробка проєктів на базі Arduino та STM32; – протоколи комунікації для IoT пристроїв.
Мета (Чому це цікаво/потрібно вивчати)	Програмування мікропроцесорної техніки є ключовою навичкою в епоху Інтернету речей (IoT), робототехніки та автоматизації. Мікроконтролери використовуються у побутовій техніці, автомобілях, медичному обладнанні, промисловій автоматизації, смартфонах та мільйонах інших пристроїв. Вивчення цієї дисципліни дає розуміння того, як програмне забезпечення взаємодіє з апаратним забезпеченням на найнижчому рівні, що є фундаментальним для створення ефективних embedded рішень. Знання програмування мікроконтролерів відкриває можливості для роботи у галузі IoT, робототехніки, автомобільної промисловості, розумних будинків, носимої електроніки та промислової автоматизації. Це затребувана спеціалізація з постійно зростаючим попитом на ринку праці.

Програмні результати <i>(Чому можна навчитися)</i>	– програмувати мікроконтролери на мовах C/C++ та асемблері; – налаштовувати GPIO порти для введення-виведення; – працювати з таймерами та генерувати ШІМ сигнали; – обробляти переривання від різних джерел; – зчитувати дані з аналогових датчиків через АЦП; – використовувати послідовні інтерфейси (UART, SPI, I2C) для комунікації; – підключати та керувати периферійними пристроями (дисплеї, датчики, двигуни); – створювати прошивки для мікроконтролерів Arduino та STM32; – налагоджувати код за допомогою debugger та логічних аналізаторів; – розробляти багатозадачні програми з використанням RTOS; – читати datasheet та reference manual мікроконтролерів; – проектувати архітектуру embedded додатків; – інтегрувати мікроконтролери з бездротовими модулями (Wi-Fi, Bluetooth); – створювати IoT пристрої з підключенням до хмарних сервісів; – тестувати та верифікувати роботу вбудованих систем; – документувати embedded проекти.
Компетентності <i>(Як можна користуватися набутими знаннями і вміннями)</i>	Дисципліна формує практичні компетенції embedded розробника: розробка вбудованого програмного забезпечення; проектування та реалізація IoT пристроїв; створення систем автоматизації та керування; розробка робототехнічних систем; програмування носимої електроніки; оптимізація продуктивності та енергоефективності; інтеграція різномірних систем та протоколів; проектування архітектури embedded рішень; технічна підтримка та супровід embedded систем.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та методичні матеріали лабораторних робіт; підручники з мікропроцесорної техніки та embedded систем; datasheet та reference manual для мікроконтролерів STM32, Arduino, ESP32.
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Мова програмування Assembler для мікропроцесорів x86
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики.
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	6 кредитів ЄКТС: – лекції – 28 годин, – лабораторні заняття – 32 години, – самостійна робота – 120 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Знання основ програмування на мовах високого рівня (C, C++, Pascal). Розуміння систем числення (двійкова, вісімкова, шістнадцяткова) та арифметичних операцій у них. Базові знання архітектури комп'ютерних систем та організації пам'яті. Знання булевої алгебри та логічних операцій.
Предмет навчання (Що буде вивчатися)	– архітектура процесорів сімейства x86/x64 (Intel, AMD); – система команд процесора та формати інструкцій; – регістри загального призначення та спеціального призначення; – режими адресації пам'яті; – сегментна та плоска моделі пам'яті; – команди обробки даних (арифметичні, логічні, бітові операції); – команди передачі керування (умовні та безумовні переходи); – робота зі стеком та підпрограмами; – організація циклів та умовних конструкцій; – робота з рядками та масивами; – інструкції роботи з числами з плаваючою комою (FPU, SSE, AVX); – взаємодія асемблера з мовами високого рівня; – системні виклики та взаємодія з операційною системою; – відлагодження асемблерних програм.
Мета (Чому це цікаво/потрібно вивчати)	Програмування на мові асемблера дає найглибше розуміння того, як комп'ютер виконує інструкції на апаратному рівні, що є фундаментальним знанням для будь-якого серйозного програміста. Знання асемблера необхідне для розробки критичних за продуктивністю фрагментів коду, драйверів пристроїв, операційних систем, вбудованих систем та низькорівневих утиліт. Це критично важливо для фахівців з кібербезпеки, reverse engineering, аналізу шкідливого програмного забезпечення та розробки антивірусів. Вивчення асемблера розвиває глибоке розуміння архітектури комп'ютера, дозволяє писати максимально оптимізований код та усвідомлювати, що насправді відбувається під час виконання програм на мовах

	високого рівня.
Програмні результати <i>(Чому можна навчитися)</i>	– писати програми на мові асемблера з використанням NASM, MASM або FASM; – використовувати всі групи команд процесора (арифметичні, логічні, бітові, переходів); – працювати з різними режимами адресації; – організовувати роботу зі стеком та передачу параметрів; – створювати та викликати підпрограми; – реалізовувати цикли та умовні конструкції; – обробляти масиви та рядки даних; – використовувати макрокоманди для спрощення коду; – використовувати системні виклики операційної системи; – працювати з введенням-виведенням; – відлагоджувати програми за допомогою debugger (GDB, OllyDbg, x64dbg).
Компетентності <i>(Як можна користуватися набутими знаннями і вміннями)</i>	Дисципліна формує компетенції низькорівневого програміста: розробка високопродуктивного коду для критичних секцій; створення драйверів пристроїв та системного ПЗ; програмування вбудованих систем та мікроконтролерів; оптимізація існуючого програмного забезпечення; аналіз вразливостей програмного забезпечення.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та методичні матеріали лабораторних робіт; довідники по інструкціях Intel та AMD; документація Intel Software Developer Manual; навчальні посібники з NASM, MASM, FASM; приклади коду та готові рішення типових задач; відеокурси з асемблера; онлайн-ресурси та форуми низькорівневого програмування; статті з оптимізації та reverse engineering; документація системних викликів Linux/Windows.
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Тестування програмного забезпечення
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики.
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	3 кредити ЄКТС: – лекції – 10 годин, – лабораторні заняття – 20 годин, – самостійна робота – 60 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Знання основ програмування на будь-якій мові (Python, Java). Розуміння життєвого циклу розробки програмного забезпечення. Базові знання про структури даних, алгоритми та принципи ООП. Вміння працювати з системами контролю версій (Git). Розуміння основних концепцій баз даних та веб-технологій. Знання принципів роботи операційних систем та командного рядка.
Предмет навчання (Що буде вивчатися)	– основи теорії тестування та забезпечення якості ПЗ; – методології тестування (Waterfall, Agile, Scrum); – рівні тестування (модульне, інтеграційне, системне, приймальне); – види тестування (функціональне, нефункціональне, регресійне); – автоматизація тестування та фреймворки (Selenium, Pytest, JUnit); – навантажувальне та стресове тестування; – дефект-менеджмент та системи відстеження помилок (Jira, Bugzilla); – метрики якості та звітність; – тестова документація (тест-плани, тест-кейси, чек-листи); – інструменти тестування та best practices.
Мета (Чому це цікаво/потрібно вивчати)	Тестування програмного забезпечення є критично важливим етапом розробки, що забезпечує якість, надійність та безпеку програмних продуктів. У сучасній IT-індустрії професія тестувальника (QA Engineer) є однією з найбільш затребуваних і служить відмінним стартом для кар'єри в технологіях. Вивчення цієї дисципліни дає розуміння того, як знаходити помилки до того, як вони потраплять до користувачів, як забезпечити стабільність систем та як автоматизувати процес перевірки якості. Знання тестування відкриває можливості для роботи як Manual QA, Automation QA Engineer, Performance Engineer, Security Tester, а також є цінним для розробників, які прагнуть писати більш якісний код.
Програмні результати (Чому можна навчитися)	– розробляти тест-плани, тест-стратегії та тест-кейси; – виконувати функціональне та нефункціональне

	тестування; – писати автоматизовані тести (Selenium WebDriver, Pytest, JUnit, TestNG); – використовувати системи управління тестами (TestRail, Zephyr); – виконувати регресійне тестування; – проводити smoke та sanity тестування; – документувати результати тестування та створювати звіти; – аналізувати метрики покриття коду.
Компетентності <i>(Як можна користуватися набутими знаннями і уміннями)</i>	Дисципліна формує комплексні компетенції QA інженера: забезпечення якості програмного забезпечення; розробка та виконання тестових сценаріїв; автоматизація тестових процесів; інтеграція тестування в процес розробки; аналіз та документування дефектів; оптимізація процесів тестування; оцінка ризиків та планування тестування; менторство junior тестувальників; технічне лідерство в QA команді; впровадження best practices тестування; аналіз метрик якості та звітність; консультування з питань якості ПЗ; участь в code review з точки зору тестованості; розробка тестової інфраструктури.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та методичні матеріали лабораторних робіт; документація інструментів тестування (Selenium, Pytest, JUnit); онлайн-курси та відеоматеріали; приклади тестової документації; Case Studies проєктів з автоматизації.
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік

Назва дисципліни	Обчислювальна геометрія та комп'ютерна графіка
Кафедра, яка забезпечує викладання	Прикладної математики та інформатики
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	3 кредити ЄКТС: – лекції – 10 годин, – лабораторні заняття – 20 годин, – самостійна робота – 60 годин
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Курс базується на поняттях, що вивчаються в дисциплінах: – математичний аналіз; – теорія ймовірностей та математична статистика; – основи програмування.
Предмет навчання (Що буде вивчатися)	Найпростіші алгоритми комп'ютерної графіки: використання комп'ютерної графіки в Python. Побудова прямих, кола. Елементарні перетворення на площині. Проекції. Відсікання невидимих об'єктів. Елементи обчислювальної геометрії: структури даних, що використовуються. Пошук випуклої оболонки. Створення зіркового полігону. Побудова зіркового полігону. Визначення належності точки. Відсікання ліній: алгоритм Цируса-Бека. Відсікання полігону: алгоритм Сазерленда-Ходжмана. Тріангуляція монотонних полігонів.
Мета (Чому це цікаво/потрібно вивчати)	отримання студентами знань, вмінь та навичок необхідних для моделювання кривих та поверхонь, розв'язання комбінаторних задач обчислювальної геометрії, машинної графіки.
Програмні результати (Чому можна навчитися)	Після вивчення даного курсу студент повинен знати: – алгоритми покрокового введення; – алгоритми Цируса-Бека та Сазерленда-Ходжмана; – алгоритми покрокової вибірки; – можливості бібліотеки PyGame.
Компетентності (Як можна користуватися набутими знаннями і вміннями)	Після вивчення даного курсу студент повинен вміти: – реалізовувати алгоритм побудови зіркового полігону; – реалізовувати алгоритм приналежності точки; – реалізовувати алгоритм відсікання; – будувати випуклу оболонку методом "обгортки подарунка"; – користуватися бібліотекою PyGame.
Інформаційне забезпечення дисципліни	Силабус дисципліни; рейтингова система оцінювання; презентації лекцій та матеріали практичних завдань,

	навчальні посібники
Матеріально-технічне забезпечення	Аудиторія теоретичного навчання, комп'ютерний клас для виконання лабораторних робіт, ноутбук, проектор, навчальна та наукова література, презентаційні матеріали
Вид семестрового контролю	Залік