



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Факультетський КАТАЛОГ

**вибіркових навчальних дисциплін
першого (бакалаврського) рівня вищої освіти
для освітніх програм спеціальностей:
121 Інженерія програмного забезпечення
123 Комп'ютерна інженерія
126 Інформаційні системи та технології**

Ухвалено на засіданні
Методичної ради
КПІ ім. Ігоря Сікорського
від 27.01.2022 р.
протокол № 3

Київ 2022

ЗМІСТ

Інструкція користувачам каталогу	7
Анотації вибіркових дисциплін	8
2 курс 4 семестр.....	8
<i>Освітній компонент 1Ф-каталогу</i>	8
Лінукоподібні операційні системи	8
Емпіричні методи програмної інженерії.....	9
Комп'ютерна графіка та мультимедіа.....	9
Основи клієнтської розробки	10
Прикладні задачі машинного навчання	10
Вступ до розробки програмного забезпечення	11
Основи розробки програмного забезпечення на платформі Microsoft.NET	12
Основи розробки програмного забезпечення на платформі Java.....	13
Основи розробки програмного забезпечення на платформі Node.JS	13
Практикум з Linux.....	14
3 курс 5 семестр.....	15
<i>Освітній компонент 2Ф-каталогу</i>	15
Системи масового обслуговування	15
Нечітка логіка	15
Програмування промислових контролерів.....	16
Мова програмування Java.....	17
Системне програмування C і C++	18
Сучасні технології розробки WEB-застосувань на платформі Microsoft.NET	18
Сучасні технології розробки WEB-застосувань на платформі Java	19
Сучасні технології розробки WEB-застосувань на платформі Node.JS.....	20
AGILE методологія розробки програмного забезпечення.....	22
Вступ до технології Data Science	22
<i>Освітній компонент 3Ф-каталогу.....</i>	24
Аналіз даних з використанням мови Python.....	24
Software development lifecycle (авторський курс компанії Неткрекер)	25
Методи та технології штучного інтелекту.....	25
Комп'ютерна графіка та обробка зображень.....	26
Функціональне програмування.....	27
Розробка мікросервісів	28
Комп'ютерна графіка та мультимедіа.....	29

Технології програмування користувацьких інтерфейсів (Front-end).....	30
Технології розроблення серверного програмного забезпечення (Back-end)	31
Технології програмування на C/Embedded.....	32
Освітній компонент 4Ф-каталогу	34
Технології розробки та просування сайтів	34
Основи Front-end технологій.....	34
Розроблення застосунків з використанням Spring Framework	35
Технологія блокчейн	36
Linux.....	36
Розробка мобільних застосунків під iOS.....	37
Розробка мобільних застосунків під Android	38
DevOps Fundamentals (авторський курс компанії Genesis)	38
Вступ до штучного інтелекту.....	39
Графічне та геометричне моделювання і створення реалістичних зображень...	40
Освітній компонент 5.1Ф-каталогу (для кафедри ОТ)	41
Основи розробки програмного забезпечення на платформі Java.....	41
Основи розробки програмного забезпечення на платформі Node.js.....	41
Розробка системних програм	42
3 курс 6 семестр	43
Освітній компонент 5Ф-каталогу	43
Основи розроблення програмного забезпечення на платформі Node.js.....	43
Розробка мобільних застосунків під Android	43
Основи Back-end технологій	44
WEB – орієнтовані технології. Основи Frontend та Backend розробок.....	45
Функціональне програмування.....	47
Мультипарадигменне програмування.....	48
Функційне програмування мовою Haskell.....	49
Технології програмування на ПЛІС (FPGA)	49
Системи безпеки програм і даних	51
Освітній компонент 6Ф-каталогу	52
Штучний інтелект в задачах обробки зображень	52
Моделі і методи штучного інтелекту	53
Основи візуального програмування	53
Протоколи промислових мереж.....	54
Робота з даними в хмарних середовищах.....	54

Сучасні операційні системи	55
Основи розробки програмного забезпечення на платформі Golang	56
Методи та системи штучного інтелекту	56
Управління IT-інфраструктурними проектами	57
Технології програмування на мові Python	59
Освітній компонент 7Ф-каталогу	60
Обробка та аналіз текстових даних на Python	60
Криміналістика в програмно-орієнтованих системах	60
Бази даних безпілотних систем та автономної робототехніки	61
Сучасні технології розробки WEB-застосувань на платформі Django (Python)..	62
Комп'ютерна графіка та обробка зображень	63
Комп'ютерна графіка та мультимедіа	64
Розробка ігрових застосувань (авторський курс компанії Genesis)	65
Розподілені інформаційні системи	65
Паралельні та розподілені обчислення	66
Гібридні комп'ютерні системи	67
Освітній компонент 8Ф-каталогу	68
Технології та протоколи мультисервісних мереж	68
Програмні засоби проектування та реалізації нейромережових систем	69
Професійне використання SQL та PL/SQL на прикладі РСУБД Oracle 11g	70
Розробка мікросервісів	71
Основи PHP (авторський курс компанії Genesis)	71
Розподілені системи та обчислення	72
Цифрова обробка сигналів	73
Математичні основи захисту даних та інформаційної безпеки	73
Технології Computer Vision	74
4 курс 7 семестр	76
Освітній компонент 9Ф-каталогу	76
Основи WEB – технологій	76
Безпроводні технології для управління смарт-середовищами	77
Розробка ігрових застосувань. Unity рішення	78
Розробка ігрових застосувань	78
Ігровий штучний інтелект	79
Створення інформаційних систем (IC) на базі Low-code платформ	80
Мобільні комп'ютерні мережі	81

Основи комп'ютерного моделювання.....	81
Освітній компонент 10Ф-каталогу	82
Імітаційне моделювання.....	82
Протоколи і алгоритми електронного голосування.....	83
Реактивне програмування.....	84
Мережеве управління та протоколи.....	85
Технології DevOps (авторський курс компанії SoftServe).....	85
Побудова IP-мереж на базі обладнання Juniper	86
Системне програмування в середовищі Unix.....	87
Системи автоматизованого проектування комп'ютерних систем.....	88
Освітній компонент 11Ф-каталогу	90
Проектування мікропроцесорних систем на базі Arduino та Raspberry PI.....	90
Інфраструктура програмного забезпечення WEB-застосувань.....	90
Проектування та реалізація операційних систем.....	91
Тестування та контроль якості (QA) вбудованих систем	92
Методика та організація наукових досліджень.....	94
4 курс 8 семестр	94
Освітній компонент 12Ф-каталогу	94
Алгоритми та математичні методи захисту інформації.....	94
Кібербезпека комп'ютерних мереж	95
Управління виробництвом на базі застосування систем класу ERP.....	97
Комп'ютерна лінгвістика.....	97
Основи науки про дані.....	98
Освітній компонент 13Ф-каталогу	99
Технології штучного інтелекту.....	99
Програмування вбудованих систем на основі архітектури ARM	100
Інженерія знань.....	101
Теорія формальних граматик	102
Розподілені системи обробки інформації	103
Статистичні методи машинного навчання	103
Мережне програмування в середовищі Unix.....	104
Освітній компонент 14Ф-каталогу	105
Інтелектуальні транспортні системи	105
Оптимізація та балансування навантажень у базах даних.....	106
Інтелектуальні вбудовані (Real Time) системи	107

Інструкція користувачам каталогу

1. Кількість вибірових навчальних дисциплін, які може обрати студент визначається навчальним планом. У навчальному плані зазначається також семестр, у якому викладається вибірова дисципліна, форма семестрового контролю – залік, види та обсяги навчальних занять, обсяг (у кредитах ЄКТС) – 4 кредити.

2. Каталог вибірових дисциплін спільний для студентів усіх спеціальностей ФІОТ, тому потрібно слугуватися інформацією з розділу «Вимоги до початку вивчення» анотацій та обирати дисципліни відповідно до рівня своєї підготовки.

3. У разі неможливості формування навчальних груп нормативної чисельності для вивчення певної вибірової дисципліни, студентам надається можливість протягом визначеного терміну здійснити повторний вибір, приєднавшись до вже сформованих навчальних груп.

4. За бажанням студента, який обрав певну вибірову дисципліну, допускається його/її приєднання до групи, в якій викладається ця дисципліна в рамках іншої освітньої програми, в тому числі, на іншому факультеті. Відповідне рішення щодо дисциплін, які викладаються на факультеті, ухвалюється деканом факультету. Щодо дисциплін, які викладаються на іншому факультеті (інституті), відповідне рішення ухвалюється деканом факультету за згодою декана того факультету (директора інституту), кафедра якого забезпечує викладання цієї дисципліни.

5. Студент не може двічі обрати одну й ту ж саму навчальну дисципліну.

6. Якщо студент із поважної причини не зміг обрати дисципліни вчасно, або виявив помилку щодо свого волевиявлення, він звертається в деканат із заявою для запису на вивчення обраних ним дисциплін, надавши відповідні документи. Студент, який знехтував своїм правом вибору, буде записаний на вивчення тих дисциплін, які завідувач випускаючої кафедри вважатиме потрібними для оптимізації навчальних груп і потоків.

7. Обрані студентом навчальні дисципліни зазначаються у його індивідуальному навчальному плані.

8. Більше інформації про порядок реалізації студентами права на вільний вибір навчальних дисциплін можна знайти у відповідному Положенні про порядок реалізації здобувачами вищої освіти права на вільний вибір навчальних дисциплін.

Анотації вибіркових дисциплін

2 курс 4 семестр

Освітній компонент ІФ-каталогу

Дисципліна	Лінуксоподібні операційні системи
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисципліни «Операційні системи».
Що буде вивчатися	Архітектура і побудова Linux-подібних операційних систем, головні підсистеми, можливі алгоритми і шляхи реалізації засобів керування ресурсами. Розглядаються методи і механізми розподілу процесорного часу, взаємодії процесів, сумісного доступу до ресурсів, розподілу пам'яті. Вивчаються принципи організації введення-виведення і файлових систем.
Чому це цікаво/треба вивчати	Linux, спершу розроблений для використання окремими ентузіастами на своїх персональних комп'ютерах, пізніше, завдяки підтримці таких компаній, як IBM, Sun Microsystems, HP, Novell та інших, набув неабиякої популярності як серверна операційна система. Linux портований на велику кількість апаратних платформ. Тепер ця ОС досить успішно використовується як на мейнфреймах та суперкомп'ютерах, так і вбудована в багато інших пристроїв (смартфони, планшетні ПК, маршрутизатори комп'ютерних мереж (роутери), пристрої автоматки, системи керування телевізорами та ігровими консолями тощо). Linux все частіше встановлюється і на настільні комп'ютери.
Чому можна навчитися (результати навчання)	Студенти повинні знати: - основи побудови Linux-подібних операційних систем, їхньої архітектури, вимоги до них, історію їх розвитку і сучасні підходи до їх реалізації; - методи і алгоритми керування локальними ресурсами комп'ютера: процесором, пам'яттю, розділюваними ресурсами; - принципами реалізації файлових систем; - проблеми реалізації мережних функцій операційних систем і способи організації віддаленого виклику процедур і розподілених файлових систем; - підходи до реалізації зазначених вище механізмів у сучасних. Студенти повинні уміти: - користуватись сучасними операційними Linux-системами; - формулювати вимоги до операційної системи для вирішення певних прикладних завдань; - здійснювати базові налаштування клієнтських операційних систем.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Керувати ресурсами обчислювальної системи, взаємодії з прикладним програмним забезпеченням, а також обґрунтовано обирати операційну систему для вирішення певних завдань і грамотно її налаштувати.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни (з РСО), опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік, 2 курс 4 семестр
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Емпіричні методи програмної інженерії
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з «Вища математика», «Дискретна математика», «Теорія ймовірностей», «Основи програмування».
Що буде вивчатися	Методи та засоби емпіричних досліджень, актуальних в аспекті ПЗ, зокрема: методи вимірювання емпіричної інформації, статистична обробка та аналіз даних, дискретні й неперервні теоретичні розподіли випадкових величин (вимірюваних ознак), способи статистичного оцінювання їхніх параметрів, перевірка статистичних гіпотез, математичне дослідження типів емпіричних залежностей між вимірюваними величинами, основ кореляційного та дисперсійного аналізу, застосування емпіричних методів для аналізу продуктивності та надійності програмних систем тощо.
Чому це цікаво/треба вивчати	- Математичні методи аналізу даних широко використовують при дослідженні різноманітних систем і процесів – природних, технічних, екологічних, економічних, соціальних тощо. З огляду на це формування відповідних знань та навичок є необхідною складовою підготовки фахівців у галузі системних наук і кібернетики, інформатики та багатьох інших галузей знань.
Чому можна навчитися (результати навчання)	Студент повинен ЗНАТИ: - існуючі підходи обробки та аналізу емпіричних даних та основні методи статистичних розрахунків за допомогою сучасних математичних пакетів;. ВМІТИ: - підготувати емпіричні дані до обробки, вибрати статистичні методи для аналізу вибірки та провести інтерпретацію одержаних результатів.
Як можна користуватися набутими знаннями і уміннями (компетентності)	- Використовувати можливості обчислювальної техніки при обробці та аналізі даних.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни (з РСО), опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік, 2 курс 4 семестр
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Комп'ютерна графіка та мультимедіа
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: вища математика, основи програмування
Що буде вивчатися	Методи представлення графічних об'єктів, колірні моделі, двовимірні перетворення, тривимірні перетворення, мультимедіа технології, управління графічними об'єктами мовою C#, використовуючи GDI,

	OpenGL, програми для обробки мультимедіа інформації та інструментальні засоби створення мультимедіа продукції
Чому це цікаво/треба вивчати	Графіка має широкий спектр областей застосування
Чому можна навчитися (результати навчання)	Проводити аналіз та обирати відповідні до поставленої задачі методи та сучасні програмні середовища, використовувати основні сучасні засоби растрової й векторної графіки, створювати об'єкти двовимірної та тривимірної графіки й анімації, вільно створювати різні зображення за допомогою GDI, OpenGL та комп'ютерних програм; обробляти й зв'язувати мультимедіа інформацію
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання можна використовувати при розробці програмного забезпечення, створенні мультимедіа продукції
Інформаційне забезпечення	Силабус, конспект лекцій, методичні вказівки до виконання лабораторних робіт, презентаційні матеріали
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 2 курс 4 семестр
Викладач	ст. викладач Хмелюк М.С.

Дисципліна	Основи клієнтської розробки
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: основи програмування
Що буде вивчатися	Принципи роботи систем контролю версій (СКВ), колективна робота над проектом використовуючи СКВ, основні принципів побудови веб-сторінок за допомогою мови розмітки HTML, оформлення зовнішнього вигляду сторінок з застосуванням каскадних таблиць стилів CSS; мова JavaScript, управління елементами об'єктної моделі документа DOM та об'єктної моделі браузера BOM за допомогою JavaScript
Чому це цікаво/треба вивчати	Розробка Web-застосунків актуальна на зараз, застосування JavaScript для розробки сайтів та додатків на платформі Node JS
Чому можна навчитися (результати навчання)	Працювати з проектами використовуючи СКВ, проектувати, розробляти клієнтські Web-застосунки з використанням мов HTML, CCS, JavaScript
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання можна використовувати в Front-end розробці, розробці сайтів та додатків на платформі Node JS
Інформаційне забезпечення	Силабус, конспект лекцій, методичні вказівки до виконання лабораторних робіт, презентаційні матеріали
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 2 курс 4 семестр
Викладач	ст. викладач Хмелюк М.С.

Прикладні задачі машинного навчання

Дисципліна	
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з: математичний аналіз, лінійна алгебра, дискретної математики, будь-якої мови програмування (бажано базові знання з Python), розуміння ООП.
Що буде вивчатися	При проходженні даної дисципліни, студенти познайомляться з технологіями машинного навчання, з термінологією та основними алгоритмами машинного навчання для класифікації, зрозуміють важливість попередньої обробки даних, навчатися працювати з непоміченими даними(кластерний аналіз) та нейронними мережами (глибокими згортковими, рекурентними). На лабораторних заняттях навчатися застосовувати алгоритми AdaBoost за допомогою scikit-learn, виконувати розпізнавання зображень, реалізуєте штучну нейронну мережу з нуля(за допомогою бібліотек TensorFlow, Keras) на мові Python.
Чому це цікаво/треба вивчати	Технології машинного навчання оточують нас, на сьогоднішній день розуміння алгоритмів та застосування їх на практиці є важливим для кожної сучасної людини, зокрема для розуміння правил безпеки персональних даних в Інтернеті.
Чому можна навчитися (результати навчання)	Сучасним методам машинного навчання та вмінню застосовувати технології та алгоритми машинного навчання при розробці програмне забезпечення для різноманітних практичних задач.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Уміння розробляти програмне забезпечення з застосуванням машинного навчання є однією з вимог ІТ-компаній до своїх майбутніх працівників. Отримані знання та уміння в даному курсі можуть стати підґрунтям для обрання теми бакалаврської або магістерської роботи, для подальших наукових досліджень.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, конспект лекцій (навчальний посібник), перелік питань до екзамену, перелік завдань до екзамену, навчальний посібник до виконання лабораторних робіт, завдання до виконання лабораторних робіт, перелік питань і завдань для проведення поточного і підсумкового контролю, завдання для комплексної контрольної роботи, завдання до самостійної роботи студентів та методичні вказівки до їх виконання.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 2 курс 4 семестр
Викладач	д.т.н., професор Онищенко В.В.

Дисципліна	Вступ до розробки програмного забезпечення
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: Теорія алгоритмів, Основи програмування
Що буде вивчатися	Основні етапи розробки програмного забезпечення. Їх особливості, послідовність проходження. Ознайомлення с такими поняттями, як мета, концепція програмного проекту, предметна область, потреби користувача, вимоги та функціонал, а також їх взаємозв'язок.
Чому це цікаво/треба	Ця інформація дозволяє отримати перші практичні навички та теоретичні

вивчати	знання для розробки успішних програмних продуктів.
Чому можна навчитися (результати навчання)	Навчитися проводити аналіз предметної області, концепції, мети, потреб користувача, і на їх основі визначення основних вимог програмного забезпечення для нескладних інформаційних систем.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здатність до абстрактного мислення, аналізу та синтезу Здатність застосовувати знання у практичних ситуаціях Здатність до пошуку, оброблення та узагальнення інформації з різних джерел. Здатність аналізувати об'єкт проектування або функціонування та його предметну область.
Інформаційне забезпечення	Силабус, конспект лекцій, вказівки до виконання комп'ютерних практикумів, презентації.
Форма проведення занять	Лекції, комп'ютерні практикуми
Семестровий контроль	Залік, 2 курс 4 семестр
Викладач	к.т.н., доцент Солдатова М.О.

Дисципліна	Основи розробки програмного забезпечення на платформі Microsoft.NET
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	знання алгоритмів та структур даних, підходів до програмування (наприклад, об'єктно-орієнтоване програмування)
Що буде вивчатися	<ul style="list-style-type: none"> мова програмування C#; підходи до побудови програми (принципи об'єктно-орієнтованого проектування - SOLID, шаблони проектування – GoF, рекомендації щодо стилю написання програм) механізм роботи LINQ основні підходи для роботи з XML в C#. технології роботи з базами даних ADO.NET.
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволять виробити певне мислення до процесу розробки програм на мові C#; писати «чистий код» та створювати «читабельне» програмне забезпечення; розробляти програми, які легко супроводжувати, модифікувати, розширювати
Чому можна навчитися (результати навчання)	ефективно використовувати базові конструкції мови програмування C# при створенні програм; правильно застосовувати різні механізми мови C#; правильно обирати та застосовувати колекції при розробці програмного забезпечення для обробки наборів даних; правильно обирати та застосовувати шаблони проектування для будови архітектури програми
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> використовувати мову програмування C# для розробки програмного забезпечення; застосувати архітектурні, алгоритмічні та програмні рішення для розробки програмного забезпечення; використовувати шаблони проектування для створення об'єктів та обробки запитів від користувача
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 2 курс 4 семестр

Дисципліна	Основи розробки програмного забезпечення на платформі Java
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	знання алгоритмів та структур даних, підходів до програмування (наприклад, об'єктно-орієнтоване програмування), мови програмування Сі або Сі++
Що буде вивчатися	<ul style="list-style-type: none"> архітектура JVM (розподіл пам'яті); мова програмування Java (основні конструкції, масиви, реалізація принципів об'єктно-орієнтованого програмування, абстрактні класи чи інтерфейси); технологія програмування на мові Java - Standard Edition (вкладені класи, generics, колекції, обробка виключень, система введення/виведення даних, логування роботи програми, багатопоточність); <p>підходи до побудови програми (архітектурний шаблон MVC, принципи об'єктно-орієнтованого проектування - SOLID, шаблони проектування – GoF, рекомендації щодо стилю написання програм)</p>
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволять виробити певне мислення до процесу розробки програм на мові Java; писати «чистий код» та створювати «читабельне» програмне забезпечення; розробляти програми, які легко супроводжувати, модифікувати, розширювати
Чому можна навчитися (результати навчання)	ефективно використовувати базові конструкції мови програмування Java при створенні програм; правильно застосовувати різні механізми мови Java; правильно обирати та застосовувати колекції при розробці програмного забезпечення для обробки наборів даних; правильно обирати та застосовувати шаблони проектування для будови архітектури програми; створювати багато поточні програми
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> писати програми у стилі Java за технологією Standard Edition; налаштовувати систему логування та обробку виключень; обирати способи збереження та обробки наборів даних; налаштовувати багато поточне програми; <p>будувати архітектуру програми за MVC шаблоном та використовувати шаблони проектування для створення об'єктів та для обробки запитів від користувача</p>
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, PCO, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 2 курс 4 семестр

Дисципліна	Основи розробки програмного забезпечення на платформі Node.JS
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання програмування (курс Основи Програмування)
Що буде вивчатися?	<ul style="list-style-type: none"> Особливості мови програмування JS та плафторми Nide.js

	<ul style="list-style-type: none"> Процес організації розробки програмного забезпечення. Підходи до архітектури програмного забезпечення.
Чому це цікаво/треба вивчати?	Із посиленням тренду на перехід більшості сервісів в веб середовище зростає запит на знання веб технологій від всіх учасників ІТ індустрії
Чому можна навчитися? (результати навчання)	Створювати інтерактивні веб-сервіси, навчитися роботі в команді, освоїти навички архітектури програмного забезпечення та конкурентної взаємодії в ПЗ.
Як можна користуватися набутими знаннями і вміннями? (компетентності)	<ul style="list-style-type: none"> Створення веб додатків Створення веб серверів Знання архітектури й конкурентного програмування будуть корисними при розробці будь-якого ПЗ
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, онлайн документація та альтернативні онлайн лекції
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 2 курс 4 семестр

Дисципліна	Практикум з Linux
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові навички взаємодії з комп'ютером та роботі в інтернеті
Що буде вивчатися	<ul style="list-style-type: none"> Базові команди взаємодії з операційною системою за допомогою командного рядка (робота с файлами, папками, інсталювання стороннього ПО) Робота с системою контролю версій git Написання скриптів мовою bash для автоматизації задач різної складності Основи комп'ютерних мереж, мережева адресація та маршрутизація, активне та пасивне мережеве обладнання. Налаштування мережевого обладнання. Взаємодія з АПІ за допомогою командного рядка.
Чому це цікаво/треба вивчати	Отримані знання та набуті навички дозволять зрозуміти основи взаємодії з операційною системою за допомогою командного рядка. Написання bash скриптів дозволять автоматизувати деякі задачі, що виникають в процесі розробки та компіляції проектів. Розуміння мережі дасть можливість правильно реалізовувати комунікацію, наприклад, між клієнтський та серверним програмним забезпеченням.
Чому можна навчитися (результати навчання)	Взаємодія с ОС за допомогою командного рядка. Правильно використовувати git в проектах Створювати та використовувати bash для автоматизації процесів різної складності Розуміти основи маршрутизації, без складностей працювати з різним мережевим обладнанням. Працювати з АПІ.
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 2 курс 4 семестр
Викладач	асист. Храменко М.С.

3 курс 5 семестр

Освітній компонент з 2Ф-каталогу

Дисципліна	Системи масового обслуговування
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін «Математичний аналіз», «Диференціальні рівняння», «Програмування», «Дискретна математика», «Теорія ймовірностей та математична статистика».
Що буде вивчатися	Основні відомості та поняття з теорії масового обслуговування (математична модель системи масового обслуговування, типи систем, вхідні потоки заявок та їх математичні моделі, характеристики систем масового обслуговування та інші) та методів дослідження систем масового обслуговування різних типів.
Чому це цікаво/треба вивчати	- У теорії і практиці моделювання систем важливе місце посідають моделі систем масового обслуговування (СМО). Такі системи зустрічаються нам щоденно. Це процеси обслуговування в черзі на заправній станції, у магазині, бібліотеці, кафе, також різні служби ремонту і медичної допомоги, транспортні системи, аеропорти, вокзали тощо. Черги виникають і за потреби скористатись телефонним зв'язком або передати повідомлення по Інтернету. Більше того, будь-яке виробництво також можна подати як послідовність таких систем. Особливого значення СМО набули в інформатиці. Це системи, мережі передавання інформації, бази і банки даних.
Чому можна навчитися (результати навчання)	У результаті вивчення навчальної дисципліни студент повинен знати : основні поняття та методи теорії масового обслуговування вміти : застосовувати засвоєні теоретичні знання при розв'язанні конкретних задач.
Як можна користуватися набутими знаннями і уміннями (компетентності)	- Працювати з моделями систем масового обслуговування та застосувати отримані знання до вирішення прикладних задач.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни (з РСО), опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Нечітка логіка
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку	Базові знання з дисциплін «Програмування», «Теорія ймовірностей та

вивчення	математична статистика», «Дискретна математика», «Емпіричні методи програмної інженерії».
Що буде вивчатися	Методи та моделі теорії нечітких множин.
Чому це цікаво/треба вивчати	Розвиток нечіткої логіки йде по шляху створення систем, що потрібні великому бізнесу і військовим. Нечітка логіка застосовується при аналізі нових ринків, біржовій грі, оцінці політичних рейтингів, виборі оптимальної цінової стратегії, оцінці рівня зрілості процесів захисту інформації і т.ін. З'явилися і комерційні системи масового застосування. Зсув центра досліджень нечітких систем вбік практичних застосувань привело до постановки цілого ряду проблем, зокрема: <ul style="list-style-type: none"> - нові архітектури комп'ютерів для нечітких обчислень; - елементна база нечітких комп'ютерів і контролерів; - інструментальні засоби розробки; - інженерні методи розрахунку і розробки нечітких систем керування, тощо.
Чому можна навчитися (результати навчання)	У результаті вивчення навчальної дисципліни студент повинен знати: <ul style="list-style-type: none"> - основні поняття і закони теорії нечітких множин; - основні поняття, методи і прийоми нечіткого аналізу; - основи нечіткої логіки; вміти: <ul style="list-style-type: none"> - визначати основні об'єкти теорії нечітких множин та виконувати з ними математичні і логічні операції; - застосовувати знання фундаментальних основ, підходи і методи нечіткого аналізу в навчанні та професійній діяльності, в інтегруванні наявних знань, нарощуванні накопичених знань; - застосовувати теоретичні положення теорії нечітких множин та нечіткої логіки в розв'язуванні різноманітних практичних задач.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Розв'язуванні різноманітних практичних задач із використанням нечітких моделей та методів нечіткої логіки.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни (з РСО), опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Програмування промислових контролерів
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з <i>дискретної математики, теорії автоматичного керування, мов та технологій програмування, архітектури комп'ютерних систем</i>
Що буде вивчатися	Підходи, засоби та технології програмування інтелектуальних промислових контролерів автоматизації та систем керування
Чому це цікаво/треба вивчати	Основою сучасних інформаційно-керуючих систем є пристрої керування, зокрема інтелектуальні промислові контролери автоматизації, оволодіння програмуванням яких дозволить будувати керуючі програми в інтелектуальних інформаційних системах

Чому можна навчитися (результати навчання)	Мови програмування промислових контролерів згідно стандарту МЕК 61131-3. Володіння засобами інструментального середовища програмування промислових контролерів CoDeSys. Програмування фізичних контролерів на реальних виробничих об'єктах.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Синтез керуючих алгоритмів в інформаційно-керуючих системах. Програмна реалізація алгоритмів мовами стандарту МЕК. Розробка нових функціональних блоків для середовища програмування промислових контролерів
Інформаційне забезпечення	Силабус
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	ст. викладач Яланецький В.А.

Дисципліна	Мова програмування Java
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з вищої та дискретної математики, процедурного та об'єктно-орієнтованого програмування, теорії алгоритмів.
Що буде вивчатися	Програма курсу передбачає вивчення ядра мови Java: принципів роботи з пам'яттю, типів даних, поглиблене вивчення об'єктно-орієнтованого програмування, робота з класами, методами, виключеннями та узагальненнями, Collections API та Java I/O, лямбда-вирази, багато поточність, рефлексія, інтернаціоналізація, логування, робота з базами даних та графічним інтерфейсом. А також система керування версіями та бібліотеки для тестування. Результатами буде створення простих додатків, заснованих на об'єктно-орієнтованій парадигмі.
Чому це цікаво/треба вивчати	Даний курс орієнтований на вивчення Java як людьми з мінімальним рівнем знання програмування так і людьми, які хочуть покращити свої знання з певних нюансів мови. Після його завершення ви зможете писати програми на Java, і будете мати основу, необхідну для подальшого поглиблення своїх знань та навичок в програмуванні.
Чому можна навчитися (результати навчання)	Особливостям мови програмування Java та основоположним навичкам, необхідним інженеру програмного забезпечення для вирішення проблем у реальному світі, від розробки простих алгоритмів та додатків до тестування та налагодження програм.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> • проектувати та розробляти пакетні додатки для вирішення різноманітних задач; • користуватися технічною документацією; • проводити тестування додатків за допомогою прикладних бібліотек; • проводити розробку з використанням систем керування версіями; • проводити аналіз побудованих додатків за допомогою програмних засобів.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції та лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	PhD, асистент Орленко С.П.

Дисципліна	Системне програмування С і С++
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: Основ програмування, Бази даних (не критично, можливе одночасне вивчення).
Що буде вивчатися	Синтаксис мов програмування С, С++; операційна система Linux, операційна система (ОС) Windows; середовища розробки QT Creator; основи ядра ОС та його функції; компілятор GCC; розробка драйверів на рівні модулів ядра, бібліотеки QT.
Чому це цікаво/треба вивчати	<ol style="list-style-type: none"> 1. Демонструє типовий підхід до розробки програм з використанням мов С, С++ для ОС Windows і Linux, у тому числі на рівні ядра ОС. 2. Дає можливість самостійно розробляти прикладні і системні програми. 3. Розкриває технології створення графічних інтерфейсів на базі QT. 4. Користується попитом у роботодавців, підтримується інформаційно з боку компанії-розробника. 5. Виводить студента на початковий професійний рівень.
Чому можна навчитися (результати навчання)	<ol style="list-style-type: none"> 1. Синтаксис і типові патерни (шаблони) мов програмування С, С++. 2. Технологію розробки програм на основі мови С з GCC. 3. Способи розробки програм на мові програмування С++ з використанням QT Creator для ОС Linux і Windows. 4. Типові шаблони для мов програмування С, С++. 5. Середовища розробки QT Creator і компілятор GCC. 6. Розробка на рівня ядра ОС у тому числі розробка драйверів на рівні модуля ядра.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ol style="list-style-type: none"> 1. Програмувати і налагоджувати програми на мові С (С++) для розробки драйверів на рівні модулів ядра ОС. 2. Розробляти програмне забезпечення на мові С++ для Linux і Windows з використанням QT Creator. 3. Створювати прикладні програми і графічні інтерфейси для операційної системи Linux і Windows; 4. Налаштовувати і документувати ПЗ.
Інформаційне забезпечення	Силабус, методичні рекомендації до виконання лабораторних робіт, інформаційна підтримка у Телеграмі:(https://t.me/SystemProgC_Cplpl група відкривається на час занять) інтегроване середовище розробки QT (https://www.qt.io/product/development-tools), довідкова система QT (https://resources.qt.io/learning-hub), компілятор (https://gcc.gnu.org).
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Катін П.Ю.

Дисципліна	Сучасні технології розробки WEB-застосунків на платформі Microsoft.NET
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку	<u>Обов'язково</u> : програмування – навички процедурного та об'єктно-

вивчення	орієнтованого програмування, наприклад, на мові Сі; знання алгоритмів та структур даних. Бажано: об'єктно-орієнтоване проектування; базове знання С#; реляційні бази даних та SQL; базовий Web – основи HTML та CSS, протокол http.
Що буде вивчатися	Устрій .Net коду та синтаксичні засоби його побудови: узагальнені типи даних, делегати, події, лямбда-вирази та інш. Принципи роботи прибиральника сміття та керування пам'яттю. Технології роботи з даними: ADO.Net, Entity Framework. Багатотопоточне та асинхронне програмування та інш.
Чому це цікаво/треба вивчати	Аналіз пропозицій працевлаштування в ІТ-галузі на загальновідомих інтернет-агрегаторах DOU.ua та інш. дає змогу зрозуміти, що розробники програмного забезпечення з використанням технологій Microsoft займають одні з перших позицій щодо попиту на їх знання та вміння, до того ж, якщо є хист та бажання вивчати сучасні мови та технології програмування, С# та .Net є досить молодим та стрімко змінюючимся напрямком створення програмного забезпечення, що надає широкі та цікаві перспективи як у навчанні, так і у подальшому працевлаштуванні
Чому можна навчитися (результати навчання)	Будувати архітектуру програмних систем із компонентів зі слабкою зв'язаністю (багат шарову, мікросервісну, доменно-орієнтовану). Розробляти програмні модулі різного призначення – бізнес-логіка, компоненти доступу до даних, десктопні та web-інтерфейси користувача, компоненти багаторазового використання, розташовані як локально, так і на ресурсах глобального призначення (наприклад, nuget.org), сервіси, розгорнуті у www.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Дисципліна надає набір вмінь та компетенцій, що є мінімально необхідними для студента, який бажає в майбутньому стати С#/Net-розробником програмного забезпечення, а саме: <ul style="list-style-type: none"> - створювати програмні продукти на мові С#, що працюють на платформах сімейства .Net, - розробляти бази даних для СУБД MS SQL Server та застосовувати їх у програмних системах .Net, - створювати і розгортати web-сервіси та інші програмні компоненти для використання їх у системах з сервісноорієнтованою та мікросервісною архітектурою, - писати комп'ютерні ігри на платформі unity, - розробляти настільне та web- сучасне програмне забезпечення, - створювати компоненти та інтегрувати їх в існуючі продукти Microsoft: Office, SharePoint
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Крамар Ю.М.

Дисципліна	Сучасні технології розробки WEB-застосувань на платформі Java
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	знання мови програмування Java, шаблонів проектування, основ побудови баз даних (БД) та мови запитів SQL, основ веб-програмування – протоколів передачі даних, підходи до процесу розробки програмного забезпечення

	(життєвий цикл програмного забезпечення)
Що буде вивчатися	<ul style="list-style-type: none"> • підходи до роботи з БД (інтерфейс JDBC, DAO, транзакції, JPA, Hibernate); • побудова веб-застосунків за технологією Enterprise Edition (Servlets, Filters, JSP, JSTL); • локалізацію інтерфейсу користувача; • фреймворк Spring (Spring Data, Spring Core, Spring Security, Spring Rest, Spring Boot); • інструмент автоматичної збірки застосунку – Maven; фреймворки модульного та інтеграційного тестування – JUnit та Mockito
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволять розробляти веб-застосунки за фреймворком Spring для будь-якої предметної області з урахуванням географічної локалі; створювати модульні та інтеграційні тести для перевірки працездатності застосунків. Все це необхідно знати та вміти сучасному розробнику застосунків на мові Java
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> • створювати веб-застосунки з використанням різних підходів; • використовувати фреймворк Spring для таких функцій як авторизація та аутентифікація, доступ до даних (БД), організація обробки запитів від користувачів; • створювати слоїсту архітектуру застосунків; • покривати програмне забезпечення застосунків тестовими сценаріями; локалізувати інтерфейс користувача
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> • будувати архітектуру веб-застосунку з використанням фреймворку Spring; • організовувати інтерфейс з підстроюванням до місця розташування користувача; • налаштувати Maven для збірки та зв'язку зі сторонніми бібліотеками; налаштувати такі функції застосунку як авторизація/аутентифікація, пагінація, транзакція, валідація даних
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, PCO, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	

Дисципліна	Сучасні технології розробки WEB-застосунків на платформі Node.JS
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання програмування (курс Основи Програмування)
Що буде вивчатися	<ul style="list-style-type: none"> • Особливості мови програмування JS та плафторми Node.js • Процес організації розробки програмного забезпечення. Підходи до архітектури програмного забезпечення.
Чому це цікаво/треба вивчати	Із посиленням тренду на перехід більшості сервісів в веб середовище зростає запит на знання веб технологій від всіх учасників ІТ індустрії
Чому можна навчитися (результати навчання)	Створювати інтерактивні веб-сервіси, навчитися роботі в команді, освоїти навички архітектури програмного забезпечення та конкурентної взаємодії в ПЗ.

Як можна користуватися набутими знаннями і вміннями (компетентності)	<ul style="list-style-type: none"> • Створення веб додатків • Створення веб серверів <p>Знання архітектури й конкуретного програмування будуть корисними при розробці будь-якого ПЗ</p>
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, онлайн документація та альтернативні онлайн лекції
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	

Дисципліна	AGILE методологія розробки програмного забезпечення
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Інженерія програмного забезпечення, Програмування, Об'єктноорієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Архітектура комп'ютерів
Що буде вивчатися	Основи термінології і основних понять. Основні методології розробки ПО. Методи вибору технологій для розробки. Огляд існуючих технологічних рішень які використовуються для створення проектів Основи застосування теоретичних знань на прикладі розробки документації навчального проекту персоналізованого для кожного студента чи групи студентів за їх вибором
Чому це цікаво/треба вивчати	Існуючі потреби та тенденції на ринку праці показують що для успішної роботи програмістом потрібно не тільки вміти писати код(програмувати) також потрібне розуміння принципів розробки ПО в цілому. До 50% часу технічної співбесіди витрачається саме на питання які стосуються методологій розробки. Оскільки домінуючою методологією розробки ПО на даний час є AGILE отримати розуміння та практику розробки власного проекту з 0 дозволить достатньо підготуватися до майбутньої роботи
Чому можна навчитися (результати навчання)	Розібратися в реальному процесі розробки ПО. Підготуватися до того що буде очікувати студента після працевлаштування на реальне робоче місце
Як можна користуватися набутими знаннями і вміннями (компетентності)	Розуміти як читати та писати проектну документацію для проекту в цілому чи його окремих компонент. Обирати методології та технології для реалізації проекту чи його компонент
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	ст. викладач Шевело О.П.

Дисципліна	Вступ до технології Data Science
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Структури даних і алгоритми, Алгоритми та методи обчислень, Системне програмування, Системне програмне забезпечення, Алгоритми і структури даних, Організація баз даних, Вступ до операційної системи Linux, Програмування, Архітектура комп'ютерів, Комп'ютерні мережі, Дискретна математика, Комп'ютерна логіка, Інженерія програмного забезпечення
Що буде вивчатися	Дисципліна має на меті: надання комплексного ґрунтового теоретичного

	<p>базису та потужних практичних навичок програмної реалізації методів, математичних моделей і алгоритмів технологічних процесів Data Science (дослідження даних): дані – інформація – знання – маніпулювання знаннями - візуалізація.</p> <p>Специфіка курсу полягає у розгляді, поряд із класичними методологіями Data Science, передових авторських розробок, отриманих у ході реалізації практичних R&D проєктів.</p> <p>Теоретичні основи Data Science надаються у формі лекційних занять з обов'язковою демонстрацією розглянутих алгоритмів у формі прикладів програмного коду.</p> <p>Практичні навички застосування технологій Data Science набуваються на лабораторних заняттях. При цьому особлива увага приділяється процесам інженерії програмного забезпечення.</p> <p>Практична частина дисципліни орієнтована на застосування мови програмування високого рівня Python з вивченням функціоналу бібліотек: Pandas, NumPy, Matplotlib, scikit-learn.</p> <p>В дисципліні розкривається суть технологічних процесів Data Science: обробка даних з метою отримання інформації – обробка інформації з метою виявлення знань – використання навичок на практиці - візуалізація результатів.</p> <p>Дисципліна складається із двох взаємопов'язаних блоків:</p> <p>1. Методологічні основи Data Science:</p> <p>1.1. Прикладний статистичний аналіз даних (моделі даних; статистичний аналіз характеристик експериментальної вибірки; обробка аномальних вимірів; оцінювання, екстраполяція та інтерполяція трендовими моделями - рекурентне згладжування і згладжування накопиченої вибірки; побудова нелінійних моделей експериментальних даних з використанням диференціальних перетворень);</p> <p>1.2. Багатокритеріальні методи прийняття рішень (багатокритеріальне оцінювання; багатокритеріальна ідентифікація; багатокритеріальний розподіл ресурсів; багатокритеріальний структурно-параметричний синтез систем);</p> <p>1.4. Інтелектуальний аналіз даних. (Технології: OLAP, Data Mining, Text Mining, Image Mining, Knowledge discovery, Speech and language recognition);</p> <p>1.3. Застосування штучного інтелекту для технологій Data Science (штучні нейронні мережі; багатокритеріальні оптимізаційні нейронні мережі; методи моделі і алгоритми самоорганізації та ситуативного аналізу).</p> <p>2. Технологічні аспекти Data Science:</p> <p>2.1. Алгоритми та технології прогнозування динаміки змін фінансових та фондових ринків (за статистичними і альтернативними моделями);</p> <p>2.2. Алгоритми та технології визначення кредитних ризиків для банківських CRM систем (скорингові та багатокритеріальні моделі);</p> <p>2.3. Алгоритми ідентифікації поточних ситуацій для виробничих CRM систем і об'єктах критичної інфраструктури (технології багатофакторного аналізу та Computer Vision).</p>
<p>Чому це цікаво/треба вивчати</p>	<p>Опанування знань, умінь та навичок, що потребують посади: Data Scientist, Data Engineer; Data Analyst – Risk Team тощо.</p>
<p>Чому можна навчитися (результати навчання)</p>	<p>1. Знання та навички застосування і реалізації базових алгоритмів Data Science:</p> <p>прикладний статистичний аналіз даних;</p> <p>багатокритеріальні методи прийняття рішень;</p> <p>інтелектуальний аналіз даних;</p> <p>застосування штучного інтелекту для технологій Data Science;</p> <p>візуалізація результатів Data Science.</p> <p>2. Знання і навички застосування базових бібліотек Python для реалізації алгоритмів Data Science: Pandas, NumPy, Matplotlib, scikit-learn.</p>

Як можна користуватися набутими знаннями і вміннями (компетентності)	Набуті компетенції з Data Science можливо застосовувати на проектах наступних прикладних галузей і технологій: 1. Аналіз даних для задач електронної комерції; 2. Аналіз даних для промислових та інфраструктурних CRM систем; 3. Аналіз візуальних та геопросторових даних різного спрямування; 4. Аналіз даних для медичних інформаційних систем.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	д.т.н., професор Писарчук О.О.

Освітній компонент з ЗФ-каталогу

Дисципліна	Аналіз даних з використанням мови Python
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: програмування, алгоритми та структури даних, математичний аналіз, дискретна математика, теорія ймовірностей.
Що буде вивчатися	основні бібліотеки мови Python для роботи з даними (NumPy, SciPy, Matplotlib, Seaborn, Scikit-learn, Pandas) та їх використання; статистична обробка даних в SciPy та Pandas; попередня підготовка даних для аналізу в Pandas та Scikit-learn; інструменти Python для візуалізації даних; основи машинного навчання та використання його базових алгоритмів за допомогою бібліотеки Scikit-learn.
Чому це цікаво/треба вивчати	Кожна людина в своїй професійній діяльності має справу зі значними обсягами даних. Аналіз цих даних дозволяє правильно їх інтерпретувати, отримати нову інформацію, допомагає приймати рішення. Python є популярною мовою програмування, яка активно використовується та має широкий інструментарій для аналізу даних.
Чому можна навчитися (результати навчання)	вдосконалити навички роботи з Python; ефективно використовувати бібліотеки мови Python для обробки, візуалізації та аналізу даних; правильно обирати методи та алгоритми для попередньої обробки даних; правильно обирати найбільш інформативні способи представлення даних; використовувати алгоритми машинного навчання для класифікації, кластеризації, регресійного аналізу і т.д.
Як можна користуватися набутими знаннями і вміннями (компетентності)	ефективно обробляти, візуалізувати та аналізувати дані, отримані внаслідок власних експериментів та досліджень; створювати програми мовою Python для роботи з даними.
Інформаційне забезпечення	Силабус, лекції, лабораторні роботи
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	ст. викладач Тимофеева Ю.С.

Дисципліна	Software development lifecycle (авторський курс компанії Неткрекер)
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання основ програмування, базові знання основ системного аналізу
Що буде вивчатися	<ul style="list-style-type: none"> Окремі питання Бізнес Аналізу при процесі розробки рішень для телеком провайдерів – Requirements engineering, compliance with industry standards (TM Forum, ONF, MEF, 3GPP, IETF тощо) Окремі питання Software Development при процесі розробки рішень для телеком провайдерів Окремі питання Application Management при процесі надання сервісу для телеком провайдерів (managed services, change requests implementation and configuration, L1\L2\L3 troubleshooting incident and problem management, change and release management, on-line monitoring and alerting, performance analyses and improvements).
Чому це цікаво/треба вивчати	Спеціалісти із знаннями в області розробки для телеком провайдерів є особливо затребуваними на ринку праці. Особливо в період впровадження новітніх телекомунікаційних технологій (5G, LEO, IoT тощо)
Чому можна навчитися (результати навчання)	Зрозуміти бізнесові та технічні контексти при розробці програмних рішень для телеком провайдерів
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання та уміння розширюють перелік потенційних роботодавців для слухачів курсу.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, PCO, опорний конспект лекцій (презентації)
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	Володимир Ущатовський - (5) Volodymyr Uschapovsky LinkedIn

Дисципліна	Методи та технології штучного інтелекту
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити ЄКТС
Мова викладання	Українська, англійська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: Алгоритми та методи обчислень, Алгоритми і структури даних, Програмування, Дискретна математика, Інженерія програмного забезпечення
Що буде вивчатися?	Базові знання про контекст, терміни, проблеми та сучасний стан досліджень в галузі штучного інтелекту. Огляд основних принципів і підходів до вивчення систем штучного інтелекту. Огляд основних напрямів досліджень і відповідних інструментів штучного інтелекту: нечіткі системи, нейронні мережі, генетичні

	алгоритми, еволюційні методи, глибинне навчання, тощо. Огляд основних моделей і алгоритмів із прикладами їх застосування та програмно-апаратними засобами реалізації.
Чому це цікаво/треба вивчати?	Галузь штучного інтелекту останніми роками переживає епоху справжнього піднесення з огляду на бурхливий розвиток нових моделей нейронних мереж, засобів програмного забезпечення та стрімкого розвитку обчислювальних ресурсів на основі графічних (ГПУ) та тензорних (ТПУ) прискорювачів. Дослідження методів штучного інтелекту наразі демонструють найбільш темпи зростання у наукових дослідженнях та широкому практичному застосуванню у багатьох галузях оточуючого нас життя: від персональних асистентів в смартфонах до самокерованих автівок. Розробка і впровадження систем зі штучним інтелектом дозволяє автоматизувати вирішення задач, що потребують знань. Задач, вирішення яких було не доступне без застосування фахівців в певній галузі.
Чому можна навчитися? (результати навчання)	Теоретичні знання та базовий практичний досвід у застосуванні різних методів штучного інтелекту до існуючих практичних проблем в широкому колі застосувань.
Як можна користуватися набутими знаннями і уміннями? (компетентності)	Здобуті знання дозволять зрозуміти місце і роль методів штучного інтелекту у загальному контексті інформаційних технологій. Ця дисципліна є необхідним етапом для підготовки до професійного рівня оволодіння спеціалізованими знаннями та навичками в галузі штучного інтелекту, які будуть викладатися у наступних навчальних дисциплінах, які присвячені більш детальному вивченню окремих методів штучного інтелекту.
Інформаційне забезпечення	Силабус, конспект лекцій, презентаційні матеріали, методичні вказівки до виконання лабораторних робіт
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Шимкович В.М.

Дисципліна	Комп'ютерна графіка та обробка зображень
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: Вміти користуватися комп'ютером на рівні спеціалізованого користувача, вміти працювати з растровими та векторними документами, базові знання в області комп'ютерної графіки, налагодження інтерфейсу користувача в межах стандартних функцій, наданих розробником типових програм з обробки графічних зображень.
Що буде вивчатися	Програми проектно-конструкторських редакторів AutoCAD, Компас 3D та Spotlight Pro для автоматизованого проектування. Параметричні бази даних систем автоматизованого проектування. Програми моделювання технічних компонентів в галузі комп'ютеризованих систем. Операції налагодження графічного середовища AutoCAD та Компас 3D. Процедури виконання різноманітних схем та креслень інженерної комп'ютерної графіки.

	Операції налагодження периферійного обладнання для вводу та виводу графічної інформації.
Чому це цікаво/треба вивчати	Цей курс має на меті познайомити слухача з тим, як працюють програми інженерної комп'ютерної графіки. Прикладом таких програм можуть бути 2D та 3D графічні зображення в сучасних інженерних проєктах. Комп'ютерна графіка це область інформаційних технологій (ІТ), в якій використовуються, в т.ч. і засоби САПР – систем автоматизованого проєктування, без яких не можливо існування сучасного промислового виробництва, а значить і успішна кар'єра майбутнього фахівця в галузі інформаційних технологій.
Чому можна навчитися (результати навчання)	Знання: – програм проєктно-конструкторських редакторів AutoCAD, Компас 3D та Spotlight Pro для автоматизованого проєктування; – параметричних баз даних систем автоматизованого проєктування; – складу основного та периферійного обладнання АРМ – автоматизованого робочого місця проєктувальника. Вміння: – виконувати налагодження графічного середовища AutoCAD та Компас 3D; – виконувати різноманітні схеми та креслення. Досвід: – розробки фрагментів креслень функціональних модулів робототехнічних пристроїв; налагоджування периферійного обладнання для вводу та виводу графічної інформації.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Використовувати отримані знання для розробки проєктів за допомогою комп'ютерних спеціалізованих графічних пакетів, що являють собою інструменти САПР.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, Силабус, РСО, контрольні завдання, методичні рекомендації до виконання комп'ютерного практикуму. Навчальний посібник «CAD-системи та мультимедіа».
Форма проведення занять	Лекції, комп'ютерний практикум
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Поліщук М.М.

Дисципліна	Функціональне програмування
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Знання мінімум однієї мови програмування та досвід створення та відлагодження програм. Основи дискретної математики.
Що буде вивчатися	Основи термінології та основних понять функціонального програмування. Принципи роботи лямбда-числення, його математичне обґрунтування та прикладне значення для програмування. Базиси функціонального підходу : імутабельність даних, предикати, рекурсія, функції вищого порядку, анонімні функції , поняття замикання. Переваги та недоліки використання функціонального програмування в сучасному програмуванні. Поняття рекурсії і основні види рекурсії.
Чому це цікаво/треба вивчати	Отримані знання дозволять використовувати мови програмування, як інструмент для досягнення практичних цілей. Покращить розуміння мов програмування, вже відомих студентам, та відкриє нові точки для розвитку у побудові сучасних комп'ютерних програм. Розуміння використання елементів функціонального програмування в мовах програмування інших

	парадигм.
Чому можна навчитися (результати навчання)	Аналізувати, практикувати та обирати мови програмування необхідні для вирішення практичних задач. Розуміти, які підходи відображені в сучасних мовах програмування різних парадигм, та які базові методи є найкращими для застосування при вирішенні задач сучасного програмування
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> • Вивчати нові мови програмування швидше • Оцінювати можливості та елегантність мов програмування та їх частин Вибирати найбільш придатні мови програмування для виконання специфічних задач
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, контрольні завдання, тексти лекцій
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Мажара О.О.

Дисципліна	Розробка мікросервісів
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Технології розробки веб-застосувань.
Що буде вивчатися	<ul style="list-style-type: none"> - Microservices introduction - Discovery Server - API Gateway - Centralized Configuration - Security - Container Management - Cloud Hosting - Scaling up - Operational Readiness (centralized logging, metrics) - Tracing - Message Broker - Web sockets
Чому це цікаво/треба вивчати	Розробка мікро-сервісів для використання в хмарах є напрямом, який є затребуваним роботодавцем. Курс містить важливі аспекти розробки, розгортання та моніторингу мікро-сервісів.
Чому можна навчитися (результати навчання)	Розробка, розгортання, моніторинг мікро-сервісів.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання та навички можна використати при працевлаштуванні.
Інформаційне забезпечення	Навчальна та робоча програма дисципліни, РСО, практичні завдання
Форма проведення	Лекції та лабораторні (комп'ютерні практикуми)

занять	
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Сирота О.П.

Дисципліна	Комп'ютерна графіка та мультимедіа
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	навички програмування на мові високого рівня; знання лінійної алгебри, геометрії та тригонометрії; розв'язування рівнянь, похідних, інтегралів; знання векторів, матриць; вміння розв'язувати системи рівнянь.
Що буде вивчатися	<ul style="list-style-type: none"> вступ до комп'ютерної графіки; програмування 2D графіки; трансформація та відображення об'єктів 3D графіки; моделювання 3D об'єктів та знаходження видимої поверхні; освітлення та рендеринг поверхні; алгоритми трасування променів та їх застосування.
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволяють розуміти принципи створення та редагування цифрових зображень, а також комп'ютерних ігор.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> пояснювати основні концепції комп'ютерної графіки, включаючи перегляд, проекцію, перспективу, моделювання та перетворення у двох та трьох вимірах; застосовувати концепції колірних моделей, моделей освітлення та затінення, текстур, трасування променів, прихованої поверхні, згладжування та рендерингу; інтерпретувати математичні засади комп'ютерної графіки; описувати основи анімації, параметричні криві та поверхні, а також точкове освітлення; визначати типовий графічний конвеєр та застосувати методи графічного програмування для створення комп'ютерної графіки; створювати ефективні програми для вирішення задач програмування комп'ютерної графіки, включаючи 3D-трансформацію, моделювання об'єктів, колірне моделювання, освітлення, текстури та трасування променів.
Як можна користуватися набутими знаннями і вміннями (компетентності)	<ul style="list-style-type: none"> створювати цифрові зображення та інтерактивну 3D графіку; розробляти комп'ютерні ігри та додатки.
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Родіонов П.Ю.

Дисципліна	Технології програмування користувацьких інтерфейсів (Front-end)
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	<p>Базові знання з дисциплін:</p> <p>Алгоритми та методи обчислень, Системне програмування, Системне програмне забезпечення, Алгоритми і структури даних, Організація баз даних, Вступ до операційної системи Linux, Програмування, Архітектура комп'ютерів, Комп'ютерні мережі, Дискретна математика, Комп'ютерна логіка, Інженерія програмного забезпечення, Комп'ютерні системи</p>
Що буде вивчатися	<p>Дисципліна має на меті: надання ґрунтовних теоретичних знань і конкретних практичних навичок розробки Front-end (клієнтського) програмного забезпечення.</p> <p>Навчальна дисципліна спрямована на формування компетенцій з розробки Front-end частини прикладних Web-застосунків. В поєднанні із навчальною дисципліною «Технології розроблення серверного програмного забезпечення (Back-end)» (7 семестр) забезпечується набуття комплексних знань і практичних навичок проектування і розробки прикладного програмного забезпечення, що базується на сучасних і перспективних клієнт-серверних технологіях розподіленої обробки і обміну інформацією.</p> <p>Теоретична і практична складові навчальної дисципліни спрямовані на набуття компетенцій у ході реалізації конкретного проекту з розробки Front-end частини із спрощеною реалізацією Back-end складової.</p> <p>Дисципліна складається із наступних блоків:</p> <ol style="list-style-type: none"> 1. Прототипування інтерфейсу програмного додатку (основи UI/UX дизайну; принципи розробки прототипів; DOM-дерево, фреймворк Bootstrap та його аналоги. LESS та SASS препроцесори); 2. Технології програмування користувацьких інтерфейсів (HTML5; CSS5; JavaScript; TypeScript; фреймворк jQuery; фреймворк React та архітектура Flux; фреймворк Angular 6 та архітектура MVC, VueJS); 3. Angular 6 (angular CLI, архітектура, модуль, контролер, директива, вираз; фільтри; моделі і валідатори; залежності та сервіси).
Чому це цікаво/треба вивчати	Опанування знань, умінь та навичок, що потребують посади: Front-end Developer, Front-end Engineer, Front-end Team Lead, DevOps Engineer тощо.
Чому можна навчитися (результати навчання)	<ol style="list-style-type: none"> 1. Знання та навички застосування і реалізації клієнт-сервісних технологій для створення ефективних Front-end частин сучасних Web застосунків: <ul style="list-style-type: none"> прототипування інтерфейсу Front-end; технології програмування користувацьких інтерфейсів. 2. Знання і навички застосування базових Front-end – технологій: HTML5; CSS3; JavaScript; TypeScript; фреймворк jQuery; фреймворк React та архітектура Flux; фреймворк Angular 6 та архітектура MVC.
Як можна користуватися набутими знаннями і вміннями (компетентності)	<p>Набуті компетенції з розроблення Front-end програмного забезпечення можливо застосовувати на проектах наступних прикладних галузей і технологій:</p> <ol style="list-style-type: none"> 1. Інтернет-комерція; 2. Електронний документообіг; 3. CRM-системи різного прикладного спрямування, побудованих на технологіях розподілених процесах збору, зберігання, обробки і обміну інформацією.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, навчально-методичний комплекс.
Форма проведення	Лекції, лабораторні роботи.

занять	
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	ст. викладач Алещенко О.В.

Технології	Технології розроблення серверного програмного забезпечення (Back-end)
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Структури даних і алгоритми, Дискретна математика, Інженерія програмного забезпечення.
Що буде вивчатися	<p>Дисципліна має на меті: надання ґрунтовних теоретичних знань і конкретних практичних навичок розробки серверного програмного забезпечення.</p> <p>Навчальна дисципліна спрямована на формування компетентцій з розробки Back-end частини прикладних Web-застосунків. В поєднання із навчальною дисципліною «Технології програмування користувацьких інтерфейсів (Front-End)» (8 семестр) забезпечується набуття комплексних знань і практичних навичок проектування і розробки прикладного програмного забезпечення, що базується на сучасних і перспективних клієнт-серверних технологіях розподіленої обробки і обміну інформацією.</p> <p>Теоретична і практична складові навчальної дисципліни спрямовані на набуття компетенцій у ході реалізації конкретного проекту з розробки Back-end частини із спрощеною реалізацією Front-end складової.</p> <p>Дисципліна складається із наступних блоків:</p> <ol style="list-style-type: none"> 1. Методологія і технології побудови та створення клієнт-серверних (Web) застосунків (модель “клієнт-сервер”, хмарні сервери, рівень мереж, протокольний обмін та інтернет стандарти: TCP/IP, WWW, XML, JSON, HTTP, HTTPS, FTP, Telnet, MIME; URL адреси та ідентифікатори URI; основні засоби дослідження помилок (ICMP, ping, traceroute); сокети, IP і адресація портів; використання проксі-сервера; сервіси пошуку імен: DNS, whois; сервіси віддаленого доступу: Telnet, SSH, Remote Desktop, VNC); 2. Поняття API та його протоколи. Організація системи (правил) обмеження доступу та авторизації. Фреймворки та набори інструментів для розробки Back-end. Взаємодія з БД шляхом використання менеджера об'єктів (ORM - Hibernate, Doctrine, typeORM, Eloquent). Технології розробки серверного (Back-end) програмного забезпечення: PHP (Symfony, Laravel), Java (Spring Boot, Java EE), JavaScript (Express, NestJS); 3. Специфікація веб сервісів. Тестування коду (необхідність розроблення специфікації; OPENAPI3 - документація; створення схем (яких?); основи Test Driven Development - unit, integration та e2e тести (PHPUnit, JUnit, Jest); перехоплення та оброблення помилок. 4. Вступ в DevOps: складові частини DevOps – Development (Software Engineering), Operations, Quality; Assurance; налаштування серверів Apache, PHP, MySQL; інтеграція DevOps в процес розробки програмного забезпечення; DevOps and Digitalization; CI / CD development; Забезпечення високої доступності сервісів – High Availability.
Чому це цікаво/треба вивчати	Опанування знань, умінь та навичок, що потребують посади: Back-end developer; Back-end Engineer; Back-end Team Lead; DevOps Engineer тощо.
Чому можна навчитися	1. Знання та навички застосування і реалізації клієнт-серверних технологій для створення ефективних Back-end частин сучасних Web застосунків:

(результати навчання)	методологія і технології побудови та створення Back-end частин клієнт-серверних за стосунків; сучасні технології розробки серверного програмного забезпечення; тестування коду; основи реалізації DevOps завдань. 2. Знання і навички застосування базових Back-end – підходів: фреймворки, тестування, розробка API.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті компетенції з розроблення серверного програмного забезпечення (Back-end) можливо застосовувати на проектах наступних прикладних галузей і технологій: 1. Інтернет-комерція; 2. Електронний документообіг; 3. CRM-системи різного прикладного спрямування, побудованих на технологіях розподілених процесах збору, зберігання, обробки і обміну інформацією.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	д.т.н., професор Писарчук О.О.

Дисципліна	Технології програмування на C/Embedded
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Дискретна математика, Комп'ютерна логіка, Комп'ютерна арифметика, Програмування, Архітектура комп'ютерів, Вступ до ОС Linux.
Що буде вивчатися	Дисципліна зосереджена на базових основах розробки програмного забезпечення для застосувань вбудованих систем із використанням операційної системи Linux. Будуть вивчатися: – базові основи програмування на мові C; – архітектура вбудованих систем на базі Linux; – розвертання та налаштування процесорного ядра із вихідних кодів; – конфігурація та розробка програмного забезпечення для вбудованих систем процесорів на базі процесорів архітектури ARM; – конфігурація та розроблення програмного забезпечення в Linux на віртуальній машині; – прошивка плат; – використання системи контролю версій Git. Наприкінці дисципліни слухачі розробляють проєкт з власною прошивкою. Для лабораторних робіт використовуються платформи BeagleBone Black, Raspberry Pi.
Чому це цікаво/треба вивчати	Вбудовані системи та пристрої IoT потребують програмного забезпечення для функціонування та досягнення своєї робочої мети. Однак писати це програмне забезпечення з нуля для кожного типу апаратних засобів не є практичним. Найпоширеніший підхід - використання операційних систем для вирішення основної задачі управління апаратними ресурсами та надання послуг для користувацьких додатків.

	<p>Галузь застосування. Розроблення рішень для Інтернету Речей (IoT), вбудованих систем, розумних систем. Розроблення апаратних систем для автомобільної промисловості, промислової автоматизації, високопродуктивних обчислень, штучного інтелекту машинного навчання.</p> <p>Курс «Технології програмування на C/Embedded» може бути прослуханий студентами як окремий базовий курс, так і у складі сертифікатної програми «Інженерія вбудованих систем та IoT», яка запроваджена на факультеті інформатики та обчислювальної техніки з 2021 – 2022 навчального року. Детально з сертифікатною програмою можна ознайомитися на сайті кафедри обчислювальної техніки. Перелік дисциплін сертифікатної програми:</p> <ul style="list-style-type: none"> • Технології програмування C/Embedded – 5 семестр; • Управління IT-інфраструктурними проектами – 6 семестр; • Технології програмування на ПЛІС (FPGA) – 6 семестр; • Тестування та контроль якості (QA) вбудованих систем – 7 семестр; • Інтелектуальні вбудовані Real Time системи – 8 семестр. <p>В цьому курсі і в сертифікатній програмі загалом розглядаються цікаві задачі розроблені у співробітництві з менторами та тренерами компанії GlobalLogic, які також часто приймають участь в лекційних та практичних заняттях. Ми намагаємося допомогти студентам краще зрозуміти реалії сучасного ринку Embedded Systems та потреби роботодавців, подолати важкий поріг входження в Embedded та світ HW.</p> <p>Дефіцит фахівців о області Embedded Systems та IoT на сьогодні доволі значний, тому опанування дисциплін сертифікатної програми дозволить розширити можливості студентів в пошуку майбутньої роботи та підвищить їх конкурентоспроможність. З точки зору компанії GlobalLogic участь в сертифікатній програмі дозволить сформувати необхідні знання та вміння, що задовольнять її, як роботодавця.</p>
<p>Чому можна навчитися (результати навчання)</p>	<ul style="list-style-type: none"> – Розгортати та налаштувати операційну систему Linux для архітектури процесорів ARM із вихідних кодів(Linux kernel, U-Boot, BusyBox). – Програмувати в командній оболонці Bash. – Використовувати систему контролю версій Git. – Програмувати на C для ОС Linux для процесорів архітектури ARM. Програмувати власні модулі для процесорного ядра. – Виконувати тестування власного продукту на базі сучасних платформ для розробки. <p>Працювати в команді.</p>
<p>Як можна користуватися набутими знаннями і вміннями (компетентності)</p>	<p>Отримані знання достатні для того, щоб розробити архітектуру та реалізувати власний вбудований пристрій, налагодити його та налаштувати для роботи як автономно так і в інфраструктурі IoT.</p> <p>Отриманий досвід та практика достатні для того щоб продовжити самостійно професійне навчання в області розроблення нових пристроїв IoT та вбудованих систем з використанням будь якого доступного на ринку апаратури устаткування та програмного забезпечення.</p> <p>Отримані досвід, знання та практика достатні для того, щоб успішно проходити співбесіди в IT компаніях, що займаються розробленням вбудованих систем та IoT.</p>
<p>Інформаційне забезпечення</p>	<p>Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс.</p>
<p>Форма проведення занять</p>	<p>Лекції, практичні та семінарські заняття, лабораторні роботи</p>
<p>Семестровий контроль</p>	<p>Залік, 3 курс 5 семестр</p>

Викладач	асистент Каплунов А.В.
----------	------------------------

Освітній компонент з 4Ф-каталогу

Дисципліна	Технології розробки та просування сайтів
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання мов програмування, основ графіки та дизайну, роботи в мережі інтернет.
Що буде вивчатися	<p>Види інтернет додатків: Landing сторінка, інформаційний сайт, інтернет-магазин, корпоративний сайт, потал. Життєвий цикл розробки web додатків.</p> <p>Front end. Створення адаптивного дизайну сайту: Usability, Progressive Enhancement, Responsive design (Media Queiries, Flexible Media, Fluid grid), XML, CSS.</p> <p>Back end: мова програмування PHP, СУБД My SQL.</p> <p>Хостинг сайту: віртуальний хостинг, виділений сервер (VPS, VDS), colocation.</p> <p>Просування сайтів: Google Ads, SEO (внутрішня та технічна оптимізація сайту, внутрішня перелінковка, зовнішні чинники), SMM, SMO, SMP.</p>
Чому це цікаво/треба вивчати	Сучасний світ характеризується стрімким розвитком і впровадженням інтернет додатків. Практично кожна компанія повинна мати власний сайт. Тому даний курс є доволі актуальним.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> ▪ Розробляти адаптивні макети сайтів; ▪ Робити адаптивну верстку сайту; ▪ Розробляти динамічні сайти за допомогою різноманітних мов web програмування; ▪ Просувати сайт в пошукових системах за допомогою SEO оптимізації.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> ▪ для розробки та просування Landing сторінок; ▪ для розробки та просування інформаційних сайтів та інтернет - магазинів; ▪ для розробки та просування корпоративних сайтів.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс.
Форма проведення занять	Лекції, практичні заняття.
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Крилов Є.В.

Дисципліна	Основи Front-end технологій
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: Базові знання мов програмування, основ графіки та дизайну, роботи в мережі інтернет.
Що буде вивчатися	Прототип сайту (Moqups, Draftium), контроль версій GitHub, HTML5 та

	CSS3 (basic, advanced), JavaScript (basic, advanced), React.js . Розробка елементів сайту. Верстка сайту за допомогою HTML-5, CSS-3. Адаптивна верстка сайту. Розробка інтерфейсу користувача за допомогою JS та React.js.
Чому це цікаво/треба вивчати	Front end – це один із найпопулярніших напрямків у IT. Front-end developers є невід’ємними учасниками будь-якої команди, яка займається розробкою веб-сайтів, веб-застосунків або будь-яких продуктів у WEB.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> ▪ розробляти прототипи сайтів; ▪ розробляти адаптивні макети сайтів; ▪ розробляти адаптивну верстку сайту; розробляти динамічні сайти за допомогою JS та React.js;
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> ▪ для розробки статичних сторінок; ▪ для розробки адаптивних WEB-сторінок; ▪ для розробки динамічних WEB-сторінок ▪ для розробки інтерфейсу користувача за допомогою JS.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс, VSCode, GitHub, Moqups, Draftium, HTML5, CSS3, JS, React.js
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Голубєв Л.П.

Дисципліна	Розроблення застосунків з використанням Spring Framework
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з Основ програмування, ООП, Веб-програмування, Баз даних
Що буде вивчатися	Використання мови Java та Spring Framework для створення веб-застосунків та веб-сервісів
Чому це цікаво/треба вивчати	<ol style="list-style-type: none"> 1. Демонструє типові підходи до розроблення веб-застосунків та веб-сервісів. 2. Дає можливість в майбутньому як самостійно створювати невеликі стартап-проекти так і працювати в великих ІТ-компаніях. 3. Spring Framework є одним з найбільш популярних фреймворків для розроблення застосунків для платформи Java на даний час.
Чому можна навчитися (результати навчання)	<ol style="list-style-type: none"> 1. Керування залежностями між компонентами на основі принципу Inversion of Control. 2. Створення веб-застосунків та веб-сервісів на основі Spring MVC. 3. Робота з базами даних за допомогою Spring Data (JDBC, JPA). 4. Декларативне керування безпекою та транзакціями в застосунках Spring Framework. 5. Створення та конфігурування застосунків Spring Framework за допомогою Spring Boot.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ol style="list-style-type: none"> 1. Створювати веб-застосунки та веб-сервіси з використанням Spring Framework. 2. Проектувати та розробляти компоненти для ентєрпрайз-рішень з мікросервісною архітектурою. 3. Розробляти рішення, що інтегруються з зовнішніми системами.

Інформаційне забезпечення	Силабус дисципліни, РСО, методичні рекомендації до виконання лабораторних робіт
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	к.т.н., доцент Букасов М.М.

Дисципліна	Технологія блокчейн
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з мов та технологій програмування, кодування та шифрування інформації, бази даних, мережеві технології
Що буде вивчатися	Підходи, засоби та технології програмування розподілених баз даних типу блокчейн. Теоретичні засади щодо архітектури, механіки роботи систем на базі блокчейн. Токеноміка та цифровізація послуг, монетизація творчості та рейтингування соціуму, суверенна особистість та децентралізовані сертифікати.
Чому це цікаво/треба вивчати	- Тотальна цифровізація та діджиталізація бізнес-процесів продукують запити до нових компетенцій серед фахівців з інформаційних систем. Зокрема, існує дефіцит у фахівцях з децентралізованих баз даних блокчейн, спеціалістів зі сфери цифрових/віртуальних активів, смарт-контрактів та децентралізованих фінансів.
Чому можна навчитися (результати навчання)	Базові підходи, методи та технології розгортання прототипу розподіленої бази даних типу блокчейн задля підвищення якості бізнес-процесів в інформаційних системах.
Як можна користуватися набутими знаннями і вміннями (компетентності)	- Допомогати у розгортанні та супроводженні розподілених блокчейн-платформ й монетизації бізнес-процесів
Інформаційне забезпечення	Силабус
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	ст. викладач Яланецький В.А.

Дисципліна	Linux
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: операційні системи
Що буде вивчатися	Структура системи, конфігурування системи, командний рядок та його використання для роботи зі стандартними потоками, текстовими редакторами, управлінням файлами, каталогами, правами доступу, користувачами та групами, процесами, журналюванням операцій. Команди мови Shell, скрипти. Планування та планувальники задач в

	Linux, управління пакетами Linux, управління апаратним забезпеченням в Linux. Проектування, монтування файлових систем. Налаштування мережевих параметрів в Linux. Налаштування графічного середовища
Чому це цікаво/треба вивчати	Linux-системи використовуються на ПК, серверах
Чому можна навчитися (результати навчання)	Працювати з системою, використовуючи командний рядок Shell; управляти файлами, каталогами, правами доступу, користувачами та групами, процесами, журналюванням операцій в Linux; конфігурувати систему через налаштування пакетів, апаратного забезпечення, файлових систем, графічного середовища, мережі.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання можна використовувати адміністрування Linux-систем
Інформаційне забезпечення	Силабус, конспект лекцій, методичні вказівки до виконання лабораторних робіт, презентаційні матеріали
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	ст. викладач Хмелюк М.С.

Дисципліна	Розробка мобільних застосунків під iOS
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	знання мови програмування C# та XML, основ побудови баз даних (БД) та мови запитів SQL, основ веб-програмування, алгоритмів та структур даних
Що буде вивчатися	архітектура операційної системи iOS та структури застосунку під неї; мова програмування Swift; розробка нативних застосунків під iOS; взаємодія з базами даних та файловою системою; доступ до глобальної мережі та розробка веб-застосунку; організація роботи з мультимедіа, датчиками, анімацією
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволять розробляти мобільні застосунки під iOS як нативні, так і гібридні застосунки для будь-якої предметної області
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> розробляти нативні застосунки під iOS; розробляти гібридні застосунки під iOS; організувати сховища для збереження даних у пам'яті пристрою або у БД; використовувати інтернет-джерела для доступу к даним
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> аналізувати та обирати тип мобільного застосунку; розробляти мобільні версії сайтів; застосовувати різні інтерактивні елементи для зручного інтерфейсу мобільного застосунку; аналізувати та обирати сховища для збереження даних з урахуванням безпеки по доступу
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий	Залік, 3 курс 5 семестр

контроль	
Викладач	

Дисципліна	Розробка мобільних застосунків під Android
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	знання мови програмування Java та XML, основ побудови баз даних (БД) та мови запитів SQL, основ веб-програмування, алгоритмів та структур даних
Що буде вивчатися	<ul style="list-style-type: none"> архітектура платформи Андроїд; мова програмування Kotlin; розробка нативних застосунків під платформу Андроїд; взаємодія з базами даних (SQLite) та файловою системою; доступ до глобальної мережі та розробка веб-застосунку; організація роботи з мультимедіа, датчиками, анімацією
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволять розробляти мобільні застосунки під платформу Андроїд як нативні, так і веб-застосунки та гібридні застосунки для будь-якої предметної області
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> розробляти нативні застосунки для мобільної платформи Андроїд; розробляти гібридні застосунки для мобільної платформи Андроїд; організовувати сховища для збереження даних у пам'яті пристрою або у БД; використовувати інтернет-джерела для доступу к мультимедійним даним; застосовувати акселерометри для застосунків
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> аналізувати та обирати тип мобільного застосунку; розробляти мобільні версії сайтів; застосовувати різні інтерактивні елементи для зручного інтерфейсу мобільного застосунку; аналізувати та обирати сховища для збереження даних з урахуванням безпеки по доступу
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	

Дисципліна	DevOps Fundamentals (авторський курс компанії Genesis)
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	<ul style="list-style-type: none"> знайомство з основами мережевих технологій базові знання з програмування ООП основи баз даних
Що буде вивчатися	<ul style="list-style-type: none"> робота з ОС Linux Networking IaC, CI/CD

	– SDLC автоматизація тестування
Чому це цікаво/треба вивчати	На сьогодні більшість нових програмних продуктів працюють в хмарах і, замість написання додатків “з нуля”, спеціалісти застосовують мікросервісну архітектуру, хмарні сервіси, і використовують вже готові програмні модулі для вирішення конкретних задач. DevOps-інженер – це “конструктор”, який збирає деталі продукту в єдине ціле.
Чому можна навчитися (результати навчання)	Освоїти методологію DevOps: адмініструвати Linux, конфігурувати та розгортати інфраструктуру, налаштовувати веб-сервери, автоматизувати всі процеси та інше.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Адмініструвати Linux, конфігурувати та розгортати інфраструктуру, налаштовувати веб-сервери, автоматизувати всі процеси та інше.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	Авторський курс компанії Genesis

Дисципліна	Вступ до штучного інтелекту
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська, англійська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Алгоритми та методи обчислень, Системне програмування, Системне програмне забезпечення, Алгоритми і структури даних, Організація баз даних, Програмування, Архітектура комп'ютерів, Комп'ютерні мережі, Дискретна математика, Інженерія програмного забезпечення, Паралельне програмування
Що буде вивчатися	Базові знання про контекст, терміни, проблеми та сучасний стан досліджень в галузі штучного інтелекту. Огляд основних принципів і підходів до вивчення систем штучного інтелекту. Огляд основних напрямів досліджень і відповідних інструментів штучного інтелекту: генетичні алгоритми, еволюційні методи, машинне навчання, глибинне навчання, навчання із підкріпленням, тощо. Огляд основних моделей і алгоритмів із прикладами їх застосування.
Чому це цікаво/треба вивчати	Галузь штучного інтелекту останніми роками переживає епоху справжнього піднесення з огляду на бурхливий розвиток нових моделей нейронних мереж, засобів програмного забезпечення та стрімкого розвитку обчислювальних ресурсів на основі графічних (ГПУ) та тензорних (ТПУ) прискорювачів. Дослідження методів штучного інтелекту наразі демонструють найбільш темпи зростання у наукових дослідженнях та широкому практичному застосуванню у багатьох галузях оточуючого нас життя: від персональних асистентів в смартфонах до самокерованих автівок.
Чому можна навчитися (результати)	Теоретичні знання та базовий практичний досвід у застосуванні різних методів штучного інтелекту до існуючих практичних проблем в широкому

навчання)	колі застосувань.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здобуті знання дозволять зрозуміти місце і роль методів штучного інтелекту у загальному контексті інформаційних технологій. Ця дисципліна є необхідним етапом для підготовки до професійного рівня оволодіння спеціалізованими знаннями та навичками в галузі штучного інтелекту, які будуть викладатися у наступних навчальних дисциплінах, які присвячені більш детальному вивченню окремих методів штучного інтелекту.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	д.т.н., професор Гордієнко Ю.Г.

Дисципліна	Графічне та геометричне моделювання і створення реалістичних зображень
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з вищої математики, фізики, аналітичної геометрії, алгоритмів та структур даних, програмування, алгоритмів та структур даних, архітектури комп'ютера
Що буде вивчатися	Моделювання просторових об'єктів і сцен за допомогою сучасних програмних засобів. Методи побудови реалістичних зображень з урахуванням законів оптики. Сучасні апаратно-програмні засоби для моделювання тривимірної реальності.
Чому це цікаво/треба вивчати	Безперервне зростання потужностей комп'ютерних засобів дозволяє використовувати якісну графіку в усе більших застосуваннях. Широко застосовуються реалістичні зображення у кіно, комп'ютерних іграх, архітектурних та геоінформаційних системах тощо. Набувають подальшого розвитку системи віртуальної реальності.
Чому можна навчитися (результати навчання)	Навчитися роботі з популярними програмними засобами комп'ютерної графіки. Вивчити популярні графічні файлові формати. Отримати знання математичних основ комп'ютерної графіки, методам створення реалістичних зображень тривимірних сцен. Вивчити сучасні апаратно-програмні засоби для моделювання тривимірної реальності.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Виконувати тривимірне геометричне моделювання на основі сучасних апаратно-програмних засобів. Знання математичних та алгоритмічних основ комп'ютерної графіки разом із знанням сучасних програмних інтерфейсів графіки відкриває можливість програмісту створювати власні програми, насичені графічними можливостями якісного відображення різноманітних об'єктів та сцен у різноманітних застосуваннях для широкого спектру ІТ-технологій.
Інформаційне забезпечення	Навчальна і робоча програма дисципліни, РСО, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	д.т.н., професор Писарчук О.О. ст. викладач Пономаренко А.М.

Освітній компонент з 5.1Ф-каталогу (для кафедри ОТ)

Дисципліна	Основи розробки програмного забезпечення на платформі Java
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання основ програмування, об'єктно-орієнтованого програмування, алгоритмів та структур даних, комп'ютерної дискретної математики
Що буде вивчатися	Принципи побудови мови Java, основні конструкції, бібліотеки. Реалізація принципів об'єктно-орієнтованого програмування, модульність, робота у мережі та з базами даних.. Класи, інтерфейси, пакети та інші основні елементи побудови програм. Засоби розробки програм та інтегровані середовища розробки та налагодження програмного забезпечення. Віртуальна машина Java. Організація багатопотокового виконання..
Чому це цікаво/треба вивчати	Оволодіння сучасними Java-технологіями створення платформонезалежного програмного забезпечення, технологіями розробки складних інформаційних систем для вирішення широкого спектру завдань.
Чому можна навчитися (результати навчання)	Створювати ефективне програмне забезпечення на основі технологій Java, вивчити та опанувати сучасні засоби розробки програм на основі об'єктно-орієнтованого підходу та популярних патернів проектування та програмування.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Здатність створювати програмне забезпечення для різноманітних операційних систем та апаратних платформ від мережних до мобільних та вбудованих.. Здатність створювати прикладне програмне забезпечення комп'ютерних систем та мереж
Інформаційне забезпечення	Навчальна і робоча програма дисципліни, РСО, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	ст. викл. Алещенко О.В.

Дисципліна	Основи розробки програмного забезпечення на платформі Node.js
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання основ програмування, об'єктно-орієнтованого програмування, алгоритмів та структур даних, комп'ютерних мереж.
Що буде вивчатися	Розробка програм мовою JavaScript. Вивчення основних принципів, базових конструкцій мови. Використання платформи Node.js для розроблення веб-систем. Засоби розробки серверних компонентів та додатків front-end.
Чому це цікаво/треба вивчати	Оволодіння сучасними технологіями розробки веб-систем створення мультиплатформного програмного забезпечення.

Чому можна навчитися (результати навчання)	Створювати ефективне програмне забезпечення веб-систем.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здатність створювати програмне забезпечення для різноманітних операційних систем та апаратних платформ. Здатність створювати прикладне програмне забезпечення комп'ютерних систем та мереж
Інформаційне забезпечення	Навчальна і робоча програма дисципліни, РСО, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік, 3 курс 5 семестр
Викладач	ст. викл. Шемсєдинов Т.Г.

Дисципліна	Розробка системних програм
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Вступ до операційної системи Linux
Що буде вивчатися	— склад системного програмного забезпечення сучасних комп'ютерних систем; — технологія покрокової розробки складних програмних продуктів Incremental Approach на прикладі розробки системних програм; — практичне застосування цієї технології під час розробки системних програм; — тестування розроблених системних програм.
Чому це цікаво/треба вивчати	— системне програмне забезпечення присутнє у будь-якому обчислювальному пристрої, тому попит на тих, хто розуміється у ньому - завжди буде високим; — системне програмне забезпечення є основою функціонування комп'ютерних систем, саме воно диктує вимоги до прикладних програм, тому його розробники будуть завжди попереду програмістів - прикладників; — розробкою системного програмного забезпечення займаються, як правило, найбільш відомі компанії, які визначають тренд розвитку комп'ютерних технологій на багато років вперед.
Чому можна навчитися (результати навчання)	— розробляти та тестувати системні програми, розуміти повідомлення операційних систем, виконувати оптимізацію коду.
Як можна користуватися набутими знаннями і уміннями (компетентності)	— при розробці та удосконаленні операційних систем, спеціалізованих програмних систем, систем комп'ютерної безпеки, систем програмування, окремих складових системного ПЗ: драйверів, програм керування файлами тощо.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота.
Семестровий	Залік, 3 курс 5 семестр

контроль	
Викладач	к.т.н., доцент Павлов В. Г.

3 курс 6 семестр

Освітній компонент з 5Ф-каталогу

Дисципліна	Основи розроблення програмного забезпечення на платформі Node.js
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з Основ програмування, Баз даних
Що буде вивчатися	Використання мови JavaScript та платформи Node.js для створення серверної частини веб-застосунків та веб-сервісів
Чому це цікаво/треба вивчати	<ol style="list-style-type: none"> 1. Демонструє типові підходи до розроблення веб-застосунків та веб-сервісів на платформі Node.js. 2. Дає можливість в майбутньому як самостійно створювати невеликі стартап-проекти так і працювати в великих ІТ-компаніях. 3. Node.js є однією з найбільш популярних платформ для розроблення серверної складової на мові JavaScript на даний час.
Чому можна навчитися (результати навчання)	<ol style="list-style-type: none"> 1. Синтаксис та основні бібліотеки мови JavaScript. 2. Принципи асинхронної багатозадачності у Node.js. 3. Використання фреймворку Express. 4. Робота з базами даних та об'єктно-реляційне відображення у застосунках на платформі Node.js.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ol style="list-style-type: none"> 1. Створювати модулі на платформі Node.js. 2. Створювати веб-застосунки та веб-сервіси за допомогою фреймворку Express. 3. Розробляти рішення, що інтегруються з зовнішніми системами за допомогою REST API.
Інформаційне забезпечення	Силабус дисципліни, PCO, методичні рекомендації до виконання лабораторних робіт
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Букасов М.М.

Дисципліна	Розробка мобільних застосунків під Android
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з теорії алгоритмів та структур даних, об'єктно-орієнтованого програмування, основ веб-програмування, роботи з базами даних та знання мов програмування Java та SQL.
Що буде вивчатися	Програма курсу передбачає:

	<ul style="list-style-type: none"> • вивчення архітектури платформи Android • розробку нативних додатків на мові Java • роботу з файловою системою та базами даних SQLite • доступ до глобальної мережі та розробки веб-додатку <p>опанування роботи з датчиками та мультимедіа</p>
Чому це цікаво/треба вивчати	Даний курс орієнтований на опанування розробки мобільних додатків під Android як людьми з мінімальним рівнем знання програмування так і людьми, які хочуть покращити свої знання з певних нюансів. Після його завершення ви зможете розробляти мобільні застосунки під платформу Android як нативні, так і веб-застосунки під ваші завдання.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> • пізнати основи розробки дизайну та функціоналу додатків під платформу Android • розробляти нативні додатки; • розробляти гібридні додатки; • організовувати сховища для збереження даних у пам'яті пристрою або у базі даних; • використовувати інтернет-джерела для доступу до мультимедійних даних; <p>оперувати датчиками для роботи з застосунками</p>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> • аналізувати та обирати тип мобільного додатку; • розробляти мобільні версії сайтів; • застосовувати різні інтерактивні елементи для зручного інтерфейсу мобільного застосунку; <p>аналізувати та обирати сховища для збереження даних з урахуванням безпеки по доступу</p>
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції та лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	PhD, асистент Орленко С.П.

Дисципліна	Основи Back-end технологій
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: Базові знання мов програмування, основ графіки та дизайну, роботи в мережі інтернет.
Що буде вивчатися	<p>Розробка серверних додатків мовою Node.js. Робота з мережними запитами. Аутентифікація та авторизація. REST API, WebSockets. Socket.IO.</p> <p>Розробка серверних додатків мовою PHP. Робота з базою даних MySQL. Підключення веб-додатків до бази даних з метою зберігання та обміну інформацією між базою даних і додатком.</p>
Чому це цікаво/треба вивчати	Back-end developer — це спеціаліст, який займається програмно-адміністративною частиною веб-додатка, внутрішнім змістом системи, серверними технологіями – базою даних, архітектурою, програмною логікою.
Чому можна навчитися	<ul style="list-style-type: none"> ▪ розробляти серверні додатки мовою Node.js;

(результати навчання)	<ul style="list-style-type: none"> ▪ розробляти серверні додатки мовою PHP; розробляти структуру БД та забезпечувати підключення веб-додатків до бази даних;
Як можна користуватися набутими знаннями і уміннями (компетентності)	При проектуванні серверної частини сайту, розробки движка сайту та серверних додатків, необхідних для роботи сайту. При проектуванні структури бази даних сайту та для організації взаємодії між базою даних та WEB-додатком.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс, VSC, GitHUB, Node.js, PHP, MySQL
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Голубев Л.П.

Дисципліна	WEB – орієнтовані технології. Основи Frontend та Backend розробок
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: «Програмування 1. Основи програмування», «Програмування - 2. Структури даних та алгоритми», «Теорія алгоритмів», «Комп'ютерні мережі», «Бази даних», «Основи клієнтської розробки», «Розроблення веб-застосунків».
Що буде вивчатися	Дана навчальна дисципліна є основою теоретичних знань, практичних вмінь та навичок розробки WEB-ресурсів та WEB-орієнтованих застосунків, які складаються з Frontend та Backend розробок. Це є основою у підготовці фахівців з інформаційних технологій для проектування, розробки, впровадження WEB -орієнтованих технологій у професійній діяльності.
Чому це цікаво/треба вивчати	Метою дисципліни «WEB-технології. Frontend та Backend розробок» є засвоєння високого рівня необхідних знань та вмінь з основ WEB – орієнтованих технологій, формування у студентів здатності проектувати, розробляти WEB-сайти та WEB-орієнтовані застосунки (Frontend та Backend розробки)
Чому можна навчитися (результати навчання)	<p>За результатами вивчення дисципліни студент повинен:</p> <p>Знати:</p> <ul style="list-style-type: none"> — структуру HTML- документу; — каскадні таблиці стилів (CSS), основні властивості CSS; — основні компоненти об'єктної моделі документа (DOM); — правила використання і основні елементи мов JavaScript та PHP. — правила використання платформ та фреймворків для Frontend та Backend розробок <p>Вміти:</p> <ul style="list-style-type: none"> — розробити структуру WEB- сайту; — застосовувати засоби каскадних таблиць стилів (CSS) для форматування сторінок WEB-сайту; — застосовувати засоби мови JavaScript при роботі з основними об'єктами HTML-документу; — застосовувати сучасні фрейворки для написання Frontend частини WEB-застосунку; — розробити базу даних для використання у динамічних WEB-

	<p>сайтах;</p> <ul style="list-style-type: none"> — використовувати платформу Node.js для написання Backend частини WEB-застосунку та зв'язку з базами даних; — використовувати засоби мови PHP для написання серверних скриптів та зв'язку з базами даних; — бути здатним вирішити складні спеціалізовані задачі та практичні проблеми у певній галузі професійної діяльності або навчання. <p>Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.</p> <p>Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.</p> <p>Демонструвати вміння аналізувати вимоги та розробляти веб-додатки, веб-сервіси, веб-сайти з використанням базових принципів, сучасних технологій та мов програмування для створення веб-застосунків, як інтерфейсу доступу до сервісів IT-інфраструктури</p>
<p>Як можна користуватися набутими знаннями і уміннями (компетентності)</p>	<p>Програмні компетентності:</p> <ul style="list-style-type: none"> — Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми в області інформаційних систем та технологій, або в процесі навчання, що характеризуються комплексністю та невизначеністю умов, які потребують застосування теорій та методів інформаційних технологій. <p>Загальні компетентності:</p> <ul style="list-style-type: none"> — Здатність до абстрактного мислення, аналізу та синтезу; — Здатність застосовувати знання у практичних ситуаціях; — Здатність до розуміння предметної області та професійної діяльності; — Здатність вчитися і оволодівати сучасними знаннями; — Здатність до пошуку, оброблення та узагальнення інформації з різних джерел; — Здатність розробляти та управляти проектами; — Здатність оцінювати та забезпечувати якість виконуваних робіт; <p>Спеціальні (фахові, предметні) компетентності:</p> <ul style="list-style-type: none"> — Здатність застосовувати технології та інструментальні засоби для розроблення веб-застосунків, веб-сервісів, веб-сайтів та веб-інтерфейсів з інтеграцією зовнішніх даних і програмних продуктів; — Здатність вирішувати задачі інтеграційних процесів інформаційних систем у сфері виробництва та керування з використанням методів аналізу та синтезу засобів передачі, зберігання та обробки інформації, основ сервіс-орієнтованого підходу до обслуговування користувачів інформаційних систем, базових та прикладних інформаційних технологій та інструментальних засобів інфраструктури IT; — Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші); — Здатність аналізувати об'єкт проектування або функціонування та його предметну область;

	<p>— Здатність управляти якістю продуктів і сервісів інформаційних систем та технологій протягом їх життєвого циклу</p> <p>— Здатність формувати нові конкурентоспроможні ідеї й реалізовувати їх у проектах (стартапах).</p> <p>Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).</p>
Інформаційне забезпечення	<p>Конспект лекцій, методичні вказівки до виконання лабораторних робіт, список основної та допоміжної літератури.</p> <p>Відповідно до технологічних вимог щодо навчально-методичного та інформаційного забезпечення освітньої діяльності відповідного рівня ВО, затверджених Постановою Кабінету Міністрів України від 30.12.2015 р. № 1187 (чинний) в редакції від 23.05.2018 р. №347.</p> <p>Університет надає доступ здобувачам до інформаційних ресурсів та електронного репозитарію Науково-технічною бібліотекою ім. Г.І. Денисенка КПІ ім. Ігоря Сікорського для організації наукових досліджень, безкоштовний доступ до інтернет-інструментарію вченого ORCID, Scopus, Web of Science тощо, авторських розробок науково-педагогічних працівників університету.</p>
Форма проведення занять	Лекції, лабораторні роботи, робота в командах
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	ст. викладач Проскура С.Л.

Дисципліна	Функціональне програмування
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Знання мінімум однієї мови програмування та досвід створення та відлагодження програм. Основи дискретної математики.
Що буде вивчатися	Основи термінології та основних понять функціонального програмування. Принципи роботи лямбда-числення, його математичне обґрунтування та прикладне значення для програмування. Базиси функціонального підходу : імутабельність даних, предикати, рекурсія, функції вищого порядку, анонімні функції , поняття замикання. Переваги та недоліки використання функціонального програмування в сучасному програмуванні. Поняття рекурсії і основні види рекурсії.
Чому це цікаво/треба вивчати	Отримані знання дозволять використовувати мови програмування, як інструмент для досягнення практичних цілей. Покращить розуміння мов програмування, вже відомих студентам, та відкриє нові точки для розвитку у побудові сучасних комп'ютерних програм. Розуміння використання елементів функціонального програмування в мовах програмування інших парадигм.
Чому можна навчитися (результати навчання)	Аналізувати, практикувати та обирати мови програмування необхідні для вирішення практичних задач. Розуміти, які підходи відображені в сучасних мовах програмування різних парадигм, та які базові методи є найкращими для застосування при вирішенні задач сучасного програмування
Як можна користуватися набутими знаннями і уміннями	<ul style="list-style-type: none"> • Вивчати нові мови програмування швидше • Оцінювати можливості та елегантність мов програмування та їх частин <p>Вибирати найбільш придатні мови програмування для виконання</p>

(компетентності)	специфічних задач
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, контрольні завдання, тексти лекцій
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Мажара О.О.

Дисципліна	Мультипарадигмне програмування
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	<ul style="list-style-type: none"> Знання мінімум однієї мови програмування та досвід створення та відлагодження програм. Основи дискретної математики.
Що буде вивчатися	<ul style="list-style-type: none"> Основи застосування методів різних парадигм програмування при проектуванні додатків Принципи використання функціонального програмування Основи функціонального програмування в Scala. Створення функцій вищого порядку в Scala Основи функціонального програмування в Clojure . Паралельне програмування в Clojure Основи функціонального програмування в Haskell. Узгодження шаблонів Функції рекурсії, контролю та вищого порядку в Haskell Основи логічного програмування в Prolog . Штучний інтелект та обчислювальна лінгвістика в Prolog
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> Логічне програмування є найбільш корисним у таких сферах, які стосуються досліджень штучного інтелекту, таких як вирішення проблем, (планування) шляху чи тлумачення природних мов. Деякі сфери, які не є строго AI, такі як парсер, також можуть значною мірою отримати користь від prolog. Однак ті, хто не працює з природними мовними системами щодня, також можуть отримати велику користь від Prolog. Оскільки Prolog використовує певні методи програмування, які вважаються складними або вдосконаленими (особливо це стосується рекурсії), Prolog може бути чудовим способом отримання розуміння цих методів. Практикуючий програміст Prolog, безумовно, буде глибоко розуміти такі поняття, як рекурсія, пошук проблемних дерев та логічне програмування обмежень. Prolog навіть може слугувати інструментом "замальовки", щоб спробувати основні ідеї програмування, перш ніж реалізувати їх на процедурних мовах. Через тісні зв'язки з логікою предикатів, вивчення Prolog може призвести до більш чіткого розуміння логіки предикатів.
Чому можна навчитися (результати навчання)	аналізувати, аргументувати, приймати рішення при розв'язанні спеціалізованих задач та практичних проблем з використанням підходи із функціональної та логічної парадигми
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> Створювати функціональні програми за допомогою Haskell Scala та Clojure Створювати логічні програми за допомогою Prolog Використовувати функціональні і логічні підходи в ООП
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, контрольні завдання, тексти лекцій
Форма проведення	Лекції та лабораторні (комп'ютерні практикуми)

занять	
Семестровий контроль	Залік, 3 курс 6 семестр

Дисципліна	Функційне програмування мовою Haskell
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання основ програмування.
Що буде вивчатися	<ul style="list-style-type: none"> теоретичні основи функційного програмування; мова програмування Haskell; інструментарій системи GHC (Glasgow Haskell Compiler); <ul style="list-style-type: none"> підходи до розробки застосувань (прикладних програм) різноманітних типів, у тому числі веб.
Чому це цікаво/треба вивчати	Тенденція до зростання кількості функційних мов та розширення функційної компоненти у складі найпоширеніших імперативних мов визначають актуальність опанування функційною парадигмою усіма розробниками програмного забезпечення в ІТ-індустрії.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> базовим поняттям функційного програмування, таким як рекурсія, каринг, зіставлення зі зразком, лінійні та енергійні обчислення, функції вищого порядку; розумінню та використанню алгебраїчних типів, класів та сортів типів; використанню функторів, аплікативних функторів та монад для організації обчислень з ефектами.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> розробляти алгоритми у функційній формі; <ul style="list-style-type: none"> створювати програмні проекти мовою Haskell з використанням GHC та Haskell Tool Stack
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	д.т.н., професор Стативка Ю.І.

Дисципліна	Технології програмування на ПЛІС (FPGA)
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з комп'ютерної логіки, комп'ютерної арифметики, дискретної математики. Базові знання архітектури комп'ютера, що включають знання загальної конструктивно-функціональної організації комп'ютерів архітектури Фон-Неймана, загальні принципи мікропрограмного керування, структуру процесорного ядра та загальні принципи організації мікропрограмного рівня. Навички програмування.

<p>Що буде вивчатися</p>	<p>Методи цифрового дизайну та проектування для FPGA. Мови опису апаратури для розроблення цифрових пристроїв. Проектування цифрових пристроїв з використанням Verilog. Функціональне моделювання та структурний синтез в сучасних САПР.</p> <p>Практичні завдання побудовані спочатку на простих прикладах програмування вузлів комп'ютерної техніки. Далі пропонується знайомство з основними етапами повного циклу розроблення IP-ядер цифрової техніки, включаючи функціональний синтез, структурний синтез, верифікацію розробок та програмування мікросхеми ПЛІС. Розглядаються базові питання створення TestBench рівнів на мові Verilog. Для перевірки правильності створених проектів та налагодження використовуються САПР ModelSym MentorGraphic, Quartus II. Розглядаються питання інтеграція САПР Quartus II и ModelSim. Для експериментів використовується налагоджувальна плати компанії Altera DE2 Board (Cyclon II), DE10 Standard Board (Cyclon IV), DE1 SoC Board (Cyclon V).</p>
<p>Чому це цікаво/треба вивчати</p>	<p>Програмована логіка (FPGA) останнім часом стає основною технологією, що використовується для створення електронних систем в різних галузях застосування. Отримані навички та знання будуть актуальні в областях розроблення рішень для Інтернету Речей (IoT), вбудованих систем, розумних систем. Розроблення апаратних систем для автомобільної промисловості, промислової автоматизації, високопродуктивних обчислень, штучного інтелекту машинного навчання.</p> <p>Курс «Технології програмування на ПЛІС (FPGA)» може бути прослуханий студентами як окремий курс, так і у складі сертифікатної програми «Інженерія вбудованих систем та IoT», яка запроваджена на факультеті інформатики та обчислювальної техніки з 2021 – 2022 навчального року. Детально з сертифікатною програмою можна ознайомитися на сайті кафедри обчислювальної техніки. Перелік дисциплін сертифікатної програми:</p> <ul style="list-style-type: none"> • Технології програмування C/Embedded – 5 семестр; • Управління IT-інфраструктурними проектами – 6 семестр; • Технології програмування на ПЛІС (FPGA) – 6 семестр; • Тестування та контроль якості (QA) вбудованих систем – 7 семестр; <p>Інтелектуальні вбудовані Real Time системи – 8 семестр.</p>
<p>Чому можна навчитися (результати навчання)</p>	<p>Проектувати цифрові схеми та пристрої з використанням сучасних методів проектування та мов проектування апаратури. Отримати навички програмування на мові Verilog. Користуватися сучасними професійними засобами для цифрового дизайну та моделювання. Виконувати функціональне моделювання і налагодження пристроїв в САПР ModelSym. Виконувати структурний синтез в САПР Quartus II Altera. Реалізувати пристрої на сучасних платах компанії Altera DE2 Board, DE10 Board, DE1 SoC Board.</p>
<p>Як можна користуватися набутими знаннями і уміннями (компетентності)</p>	<p>Базові знання у використанні мови проектування апаратури, Verilog, а також досвід і навички використання сучасних САПР і продуктів компанії Altera достатньо для продовження професійного навчання в напрямку проектування апаратури для складних систем на FPGA та ASIC, систем-на-кристалі (SoC), вбудованих систем, розумних систем. програмно-апаратних реалізацій систем штучного інтелекту.</p> <p>Отримані базові знання програмування та методів цифрового дизайну дозволять за необхідності швидко перейти на використання мови програмування апаратури VHDL, мікросхем інших виробників FPGA (Xilinx), а також інших САПР для функціонального та структурного синтезу на FPGA та ASIC (Synopsys, Aldec, Cadence).</p> <p>Отримані базові знання допоможуть у подальшому професійному навчанні в тому числі самостійно.</p>
<p>Інформаційне забезпечення</p>	<p>Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс.</p>

Форма проведення занять	Лекції, практичні та семінарські заняття, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	д.т.н., професор Клименко І.А.

Дисципліна	Системи безпеки програм і даних
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з архітектури комп'ютера, комп'ютерних мереж, алгоритмів та структур даних, основ програмування, дискретної математики, теорії ймовірності та математичної статистики.
Що буде вивчатися	Цей курс присвячений поглибленому вивченню окремих питань щодо математичних та алгоритмічних аспектів захисту інформації у комп'ютерних системах та проектуванню та розробці захищеного програмного забезпечення та систем захисту. У такому курсі вивчаються методи і алгоритми захисту при передачі та зберіганні даних, а також вивчаються питання розробки програмно-апаратних засобів захисту. Зокрема, захищені мережні протоколи, методи шифрування, криптографії, стеганографії. Питання створення та організації захищених сервісів з використанням хмарних технологій у різноманітних галузях, зокрема, електронної комерції, електронного банкінгу, функціонування криптовалют.
Чому це цікаво/треба вивчати	Сучасні реалії функціонування цифрового суспільства вимагають використання та розробки комп'ютерних засобів та систем, стійких до несанкціонованого доступу та використання програм та даних. Кожний кваліфікований фахівець, який має відношення до програмного забезпечення повинен знати принципи інформаційної безпеки та методи захисту програм і даних та вміти їх грамотно застосовувати.
Чому можна навчитися (результати навчання)	Навчитися приймати рішення щодо вибори структури засобів захисту даних; розумітися на математичних методах і алгоритмах криптографії і шифрування даних. Розробляти програмні засоби захисту інформації, стійкі до хакерських атак. Організувати захищене зберігання, обробку та передавання інформації з використанням технологій розподілених та хмарних систем.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здатність аналізувати, вибирати, розробляти і застосовувати методи і засоби для забезпечення інформаційної безпеки, кібербезпеки. Можливість виконувати роботи у якості аналітика комп'ютерних систем та програмного забезпечення. Бути кваліфікованим розробником програмного забезпечення та користувачем програмно-апаратних комплексів.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Волокита А.М.

Освітній компонент з 6Ф-каталогу

Дисципліна	Штучний інтелект в задачах обробки зображень
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з: математичний аналіз, лінійна алгебра, будь-якої мови програмування (бажано базові знання з Python), розуміння ООП.
Що буде вивчатися	При проходженні даної дисципліни, студенти познайомляться з поняттям штучного інтелекту та його розділом машинним навчанням, з термінологією та основними поняттями з обробки та аналізу цифрових зображень, з сучасними технологіями візуального покращення та відновлення зображень. Познайомляться з алгоритмами покращення та відновлення напівтонових та кольорових зображень; з морфологічною обробкою зображень. На лабораторних заняттях студенти будуть обробляти та аналізувати зображення засобами бібліотеки PythonOpenCV, виконувати перетворення зображень у різні формати, добувати корисну інформацію з зображень; використовувати традиційні технології розпізнавання образів на наборах даних(наприклад CIFAR-10). Проектувати та навчати нейронні мережі в середовищі Keras та TensorFlow для класифікації зображень
Чому це цікаво/треба вивчати	1.Вам подобається фотографувати і Вам було б цікаво навчитись обробляти та аналізувати зображення за допомогою Python? 2.Вам цікаво зрозуміти як відбувається розпізнавання людей, обличь або номерів на авто з відеокамер? 3.Вас цікавлять математичні основи вбудованих бібліотек Python для розпізнавання зображень? 4. Було б цікаво навчити нейронну мережу розпізнавати рукописний текст? Якщо хоча б на одне з питань Ви відповіли – ТАК, то даний курс саме для Вас
Чому можна навчитися (результати навчання)	Сучасними методам машинного навчання та вмінню застосовувати сучасні технології та класичні алгоритми при розробляти програмне забезпечення для обробки та аналізу зображень.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання та уміння в даному курсі можуть стати підґрунтям для обрання теми бакалаврської або магістерської роботи, для подальших наукових досліджень. Уміння розробляти програмне забезпечення для обробки та аналізу зображень можна застосувати при розробці комп'ютерних ігор, в напрямку «Розумний будинок»
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, конспект лекцій (навчальний посібник), перелік питань до екзамену, перелік завдань до екзамену, навчальний посібник до виконання лабораторних робіт, завдання до виконання лабораторних робіт, перелік питань і завдань для проведення поточного і підсумкового контролю, завдання для комплексної контрольної роботи, завдання до самостійної роботи студентів та методичні вказівки до їх виконання.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	д.т.н., професор Онищенко В.В.

Дисципліна	Моделі і методи штучного інтелекту
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: вища та дискретна математика, теорія імовірності та мат. статистика, програмування.
Що буде вивчатися	Основи штучного інтелекту, основні сучасні підходи до побудови інтелектуальних систем; огляд основних технологій штучного інтелекту: машинного навчання, класичних методів пошуку рішень та навчання з підкріпленням, моделей і методів на основі знань
Чому це цікаво/треба вивчати	Інтелектуальні технології на сьогодні широко використовуються в найрізноманітніших сферах. Але що входить в сучасний термін «штучного інтелекту»? Дисципліна дозволяє скласти комплексне уявлення про структуру та можливості інтелектуальних технологій та практично познайомитися з ними. Курс є базовим для інших спеціалізованих технологій штучного інтелекту.
Чому можна навчитися (результати навчання)	Розуміти роль, місце, тенденції та перспективи інтелектуальних систем та технологій; оволодіти основними підходами до створення моделей штучного інтелекту та технологією їх розробки; знати основні методи штучного інтелекту та сфери їх застосування, отримати навички практичного використання основних моделей та технологій штучного інтелекту.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> - вибирати раціональні методи та технології штучного інтелекту для розв'язання поставленої задачі. - застосовувати технології штучного інтелекту для вирішення практичних задач; - програмно реалізовувати методи і моделі штучного інтелекту та розробляти інтелектуальних агентів. - правильно спланувати подальше поглиблене вивчення найбільш цікавих інтелектуальних технологій
Інформаційне забезпечення	Силабус дисципліни, РСО, мультимедійні презентації та відео лекцій, комп'ютерні практикуми, Google Class
Форма проведення занять	Лекції, лабораторні роботи (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Олійник В.В.

Дисципліна	Основи візуального програмування
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з <i>вищої математики, теорії алгоритмів, класичних мов та технологій програмування</i>
Що буде вивчатися	Підходи, засоби та технології візуального програмування виконавчих платформ в задачах автоматизації інженерних досліджень. Основні можливості програмного комплексу NI LabVIEW
Чому це цікаво/треба вивчати	Альтернативний підхід до розробки програмних рішень із використанням технологій візуального програмування дозволяє надшвидкими темпами отримати готові рішення зокрема систем моніторингу в інформаційних системах, збору та обробки даних з мінімальним часом відлагодження та

	виправлення помилок.
Чому можна навчитися (результати навчання)	Базові навички візуального програмування апаратних платформ автоматизованих інформаційних систем керування збору та обробки даних засобами LabVIEW.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробляти програмні рішення в сфері автоматизації інженерних досліджень, розумний дім та автоматизованих інформаційних системах керування технологічними/виробничими процесами
Інформаційне забезпечення	Силабус
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	ст. викладач Яланецький В.А.

Дисципліна	Протоколи промислових мереж
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з <i>інформатики, дискретної математики, мов та технологій програмування, архітектури комп'ютерних систем, теорії інформації та кодування</i>
Що буде вивчатися	Огляд комунікаційних протоколів промислових мереж. Поглиблене вивчення відкритого протоколу MODBUS для комунікаційних задач промислової автоматизації та диспетчеризації інформаційних систем
Чому це цікаво/треба вивчати	Фізичні пристрої в інформаційно-керуючих системах спілкуються стандартизованими комунікаційними мовами задля запитів та отримання корисної інформації про стан інформаційної системи. Оволодіння такими мовами комунікації дозволить будувати користувацькі протоколи вбудованих та промислових мереж
Чому можна навчитися (результати навчання)	Правила формування пакетів даних в протоколі MODBUS. Обрахунок контрольних сум. Формування команд зчитування та запису даних. Програмування пакетних команд протоколом MODBUS засобами CoDeSys. Уміння тестувати та діагностувати промислову мережу MODBUS
Як можна користуватися набутими знаннями і уміннями (компетентності)	Програмування стандартизованих промислових протоколів передачі даних. Розробка нових протоколів передачі даних. Удосконалення та модифікація існуючих протоколів промислових мереж.
Інформаційне забезпечення	Силабус
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	ст. викладач Яланецький В.А.

Дисципліна	Робота з даними в хмарних середовищах
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку	Базові знання мов програмування та роботи з комп'ютером, англійська мова

вивчення	на рівні pre-intermediate. Бажання набути знань та навичок в роботі з даними з точки зору розробника та адміністратора.
Що буде вивчатися	<ul style="list-style-type: none"> • Теорія даних, робота та функції СУБД (систем управління базами даних), архітектура баз даних • SQL (structured query language) • Технології резервного копіювання та відновлення даних • Побудова звітності та візуалізація даних • Засоби забезпечення високої доступності • Хмарні середовища AWS, Azure, Google Cloud Захист даних
Чому це цікаво/треба вивчати	Збереження та обробка даних – це одні з перших задач, які виникли перед ІТ-індустрією. Будь-який спеціаліст в галузі, включаючи тестувальників, розробників, адміністраторів та архітекторів, в тій-чи іншій мірі повинні знати та вміти дістати потрібну інформацію, подати її в потрібному вигляді, впевнитись в тому, що вона не спотворена і інші користувачі та системи можуть швидко та безпечно зробити те ж саме. З впровадженням хмарних технологій, що дозволяють мінімізувати час розгортання СУБД та зробити дані доступними в будь-який час та з будь-якої точки земної кулі, подібні навички стали тільки більш актуальними.
Чому можна навчитися (результати навчання)	Буде вивчено повний цикл від проектування БД та обрання хмарного рішення для її розміщення до забезпечення високої доступності та захисту інформації.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розуміння як можна отримати потрібні дані та чи є вони вразливими. Мігрувати дані, створювати репліки, резервні копії, тощо. Писати запити та оптимізувати їх для швидкої маніпуляції даними. Проектувати бази даних з нуля для будь-яких проектів.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, контрольні завдання, навчальні матеріали.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	ст. викладач Недашківський Є.

Дисципліна	Сучасні операційні системи
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Знання основ комп'ютерних систем і мереж, алгоритмів та структур даних, підходів до програмування, засобів візуалізації інтерфейсу користувача.
Що буде вивчатися	<ul style="list-style-type: none"> • Особливості сучасних операційних систем. • Процес організації роботи операційних систем. • Підходи до архітектури операційних систем. • Практики використання можливостей операційних систем. Процес пошуку проблем швидкодії та практики оптимізації.
Чому це цікаво/треба вивчати	Внаслідок прискореного переходу більшості користувачів на мобільні комп'ютерні пристрої та все чіткішої сегментації способів використання комп'ютерної техніки в різних сферах діяльності людини зростає попит на знання актуальних операційних систем від всіх учасників ІТ-індустрії.
Чому можна навчитися (результати навчання)	Сформувати уявлення про внутрішність операційних систем та розуміння можливостей, які вони дають прикладному програмісту. Виявляти проблеми

	швидкодії та оптимізувати роботу ПЗ.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> сформувані уявлення про внутрішність операційних систем та розуміння можливостей, які вони дають прикладному програмісту. Створення оптимізованого ПЗ з урахуванням особливостей операційних систем. <p>Знання архітектури операційних систем будуть корисними при розробці широкого спектру ПЗ.</p>
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, презентації лекційного матеріалу, онлайн-документація та альтернативні онлайн лекції
Форма проведення занять	Лекції, лабораторні роботи (комп'ютерні практикуми), самостійна робота.
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	ст. викладач Ковтунець О.В.
Дисципліна	Основи розробки програмного забезпечення на платформі Golang
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання програмування (курс Основи Програмування)
Що буде вивчатися	<ul style="list-style-type: none"> Особливості мови програмування Golang Розробка програм в конкурентному середовищі Підходи до покращення продуктивності програм <p>Підходи та шаблони програмування для побудови якісної архітектури.</p>
Чому це цікаво/треба вивчати	Golang простий і потужний інструмент для розробки ПЗ. Особливо ефективний та популярний для розробки мікросервісів, а також хмарних технологій в цілому. Зокрема використовується в таких додатках як Uber, Kubernetes, Docker.
Чому можна навчитися (результати навчання)	Створювати продуктивні додатки, зокрема веб сервери.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> Створення веб серверів Створення мікросервісів <p>Розробка високопродуктивного ПЗ</p>
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, онлайн документація
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	ст. викладач Бабійчук А.

Дисципліна	Методи та системи штучного інтелекту
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Повний курс дискретної математики, повний курс вищої математики та теорії ймовірності
Що буде вивчатися	Що таке штучний інтелект : Історія ШІ , Тест Тьюрінга та чаттерботів, Розподілені, автономні та навчальні агенти, Служба робототехніки

	<p>Системи на основі знань: Логіка предикатів першого порядку, Обмеження логіки, Проблема простору пошуку, Моделювання невизначеності</p> <p>Пошук, ігри та вирішення проблем : Неінформований пошук, Евристичний пошук Ігри з супротивниками Функції евристичного оцінювання</p> <p>Обґрунтування невизначеності : Обчислення з ймовірностями , Принцип максимальної ентропії, Розум з байєсівськими мережами</p>
Чому це цікаво/треба вивчати	<p>Штучний інтелект універсальний</p> <p>ШІ не обмежується лише комп'ютерною та космічною галузями, але також відіграє значну роль у галузях, які безпосередньо пов'язані зі звичайною людиною, як сфера охорони здоров'я, автомобілебудування та навіть банківський та фінансовий сектор. Вона застосовна для будь-якої галузі, і ніколи не дає результатів.</p> <p>Майстерність століття</p> <p>Безперечно коли-небудь AI замінить багато робочих місць, а також створить багато і різних можливостей для роботи в суміжних галузях. Для того, щоб бути сучасним, кожен повинен мати хоча б якусь інформацію про AI.</p> <p>Покращене управління катастрофами</p> <p>Коли стихійне лихо вражає регіон чи державу, часто люди, які проживають там, звертаються до соціальних медіа з проханням про допомогу, окрім допомоги, яку отримують від урядових організацій. Багато разів жертви таких катастроф записують відео та діляться ними на платформах соціальних медіа, таких як Facebook та Twitter. На цих платформах є програми з підтримкою ШІ, які служать носієм для поширення новин про ці катастрофи.</p>
Чому можна навчитися (результати навчання)	Створювати системи штучного інтелекту для вирішення прикладних задач із пошуком та логічним виводом
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> • Створювати системи штучного інтелекту з нуля • Вибирати коректні підходи до створення систем штучного інтелекту <p>Проводити аналіз якості роботи систем штучного інтелекту та визначати коректність їх роботи</p>
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, контрольні завдання, підручник
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	

Дисципліна	Управління IT-інфраструктурними проектами
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	<p>Базові знання з дисциплін:</p> <p>Програмування, Архітектура комп'ютерів, Дискретна математика, Комп'ютерна логіка, Інженерія програмного забезпечення, Основи паралельного програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і</p>

	структури даних, Комп'ютерні мережі
Що буде вивчатися	<p>Головна мета дисципліни – набуття теоретичних знань і практичних навичок управління проектами та командної роботи: набуття Soft skills шляхом застосування своїх Hard skills для розв'язку задач практики.</p> <p>Реалізується у формі організації і реалізації конкретного стартап-проекту в сфері інформаційних технологій та комп'ютерної інженерії.</p> <p>Зміст дисципліни враховує найкращі практики провідних ІТ-підприємств і передбачає залучення менторів – практиків.</p> <p>Результатом опрацювання завдань дисципліни є завершений практичний проект.</p> <p>Дисципліна передбачає вивчення двох взаємопов'язаних компонент.</p> <p>I. Менеджмент проектів:</p> <p>організація та реалізація: допроектного R&D етапу; розробка технічної пропозиції і технічного завдання; інженерія вимог (бізнес-аналіз); архітектурне проектування; розробка системи (програмного забезпечення); оцінювання якості (верифікація і тестування); впровадження і основи DevOps; супроводження; організація процесів (моделей життєвого циклу) за класичними та Agile моделями; звітування і здача проектну.</p> <p>II. Групова динаміка.</p> <p>теорія і практика: професії і посади на ІТ-підприємствах та коло їх обов'язків; складання дієвого резюме; проходження співбесіди; підбір талановитого персоналу; формування групи і команди, організація командної роботи; лідерство; конфліктологія; там-менеджмент; підготовка, організація і проведення ефективних нарад, презентацій, переговорів.</p>
Чому це цікаво/треба вивчати	<p>Отриманні знання і навички забезпечують розумінні, участь і реалізацію сучасних процесів командної розробки складних програмно-апаратних систем і комплексів на рівні передових національних і світових практик.</p> <p>Курс «Управління ІТ-інфраструктурними проектами» може бути прослуханий студентами як окремий курс, так і у складі сертифікатної програми «Інженерія вбудованих систем та IoT», яка запроваджена на факультеті інформатики та обчислювальної техніки з 2021 – 2022 навчального року. Детально з сертифікатною програмою можна ознайомитися на сайті кафедри обчислювальної техніки. Перелік дисциплін сертифікатної програми:</p> <ul style="list-style-type: none"> • Технології програмування C/Embedded – 5 семестр; • Управління ІТ-інфраструктурними проектами – 6 семестр; • Технології програмування на ПЛІС (FPGA) – 6 семестр; • Тестування та контроль якості (QA) вбудованих систем – 7 семестр; <p>Інтелектуальні вбудовані Real Time системи – 8 семестр.</p>
Чому можна навчитися (результати навчання)	<p>Знання та навички організації, застосування і реалізації:</p> <ul style="list-style-type: none"> – командного виконання проектів з використанням усіх фаз життєвого циклу (від збору вимог до впровадження); – виконання проектів з використанням різних методологій розробки (класичних та Agile) та на різних ролях (розробник, тестувальник, аналітик, DevOps, керівник тощо); – використання автоматизованих засобів командної розробки проектів; – бізнес-комунікації (наради, презентації, активне слухання,

	письмова комунікація, проходження співбесіди тощо); – роботи в команді (надавання зворотного зв'язку, ділення досвідом, вирішування конфліктних ситуацій тощо); – управління часом; – практичного вирішення проблемних ситуацій; пошуку, збору та обробки інформації, необхідної для реалізації практичних завдань.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Командна робота над складними технічними проектами відповідно до сучасних вимог до процесів і технологій їх реалізації.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	д.т.н., професор Писарчук О.А.

Дисципліна	Технології програмування на мові Python
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Інженерія програмного забезпечення, Основи паралельного програмування
Що буде вивчатися	Методи перетворення проектної документації в реальний код. Способи якими оптимізації процесу написання коду для того щоб виконати поставлені задачі вчасно trade-off стратегії, робота з ризиками проекту і тд. Навички розподілення задач в часі та планування повного циклу розробки Тестування коду Розгортання коду на серверній інфраструктурі.
Чому це цікаво/треба вивчати	Однією з проблем сучасної розробки ПО є необхідність не тільки якісно писати код але й вміння витримувати строки без сильних втрат якості. Також необхідно вміти працювати як з ризиками, пріоритетами і тд.
Чому можна навчитися (результати навчання)	Отримані знання дозволять навчитися перетворювати проектну документацію в реальний код проекту дотримуючись вимог як по часу так і по якості коду. Роботі в команді розробки.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробляти ПО в команді або самостійно.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, практичні та семінарські заняття, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	ст. викладач Шевело О.П.

Освітній компонент з 7Ф-каталогу

Дисципліна	Обробка та аналіз текстових даних на Python
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: програмування, алгоритми та структури даних, математичний аналіз, дискретна математика, теорія ймовірностей. Базові знання мови Python
Що буде вивчатися	популярні бібліотеки мови Python для обробки і аналізу текстових даних, а саме: NLTK, Textblob, GenSim, SpaCy, Scikit-learn; основи роботи з текстами в Python; основні задачі обробки природної мови (NLP) та їх виконання в NLTK та Textblob; визначення тематики текстів в GenSim; машинне навчання та класифікація текстів з SpaCy та Scikit-learn.
Чому це цікаво/треба вивчати	Величезні обсяги даних представлені в текстовому неструктурованому та напівструктурованому форматі, наприклад, лог-файли, статті, відгуки користувачів, пости у соцмережах. То ж існує потреба у видобутку інформації з тексту і використанні її для визначення трендів, настроїв, побудови діалогових систем і т.д. Python є популярною мовою програмування і має низку зручних бібліотек для обробки текстових даних.
Чому можна навчитися (результати навчання)	як обробляти тексти в Python; як використовувати бібліотеки Python для попередньої обробки текстових даних; як візуалізувати текстові дані; як виконувати основні завдання обробки природної мови; як класифікувати тексти; як використовувати глибоке навчання та NLP для створення чат-ботів
Як можна користуватися набутими знаннями і уміннями (компетентності)	створювати програми мовою Python для обробки текстових даних, аналізу системних логів, патентного пошуку, класифікації документів, чат-ботів і т.д.
Інформаційне забезпечення	Силабус
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	ст. викладач Тимофєєва Ю.С.

Дисципліна	Криміналістика в програмно-орієнтованих системах
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з операційних систем, комп'ютерних мереж, безпеки бездротових і мобільних мереж (не критично, можна одночасне вивчення)
Що буде вивчатися	Основи комп'ютерної криміналістики (Forensics), основні методи та

	інструменти пошуку інформації по відкритим джерелам (OSINT), основні методи та інструменти криміналістики даних на локальній машині (Computer forensics), у мережі (Network forensics), на мобільних пристроях (Mobile device forensics), криміналістичного аналізу даних (Forensic data analysis),
Чому це цікаво/треба вивчати	<ol style="list-style-type: none"> 1. Демонструє типовий підхід до проведення розслідування з комп'ютерної криміналістики. 2. Дає можливість самостійно вивчити технології Forensics. 3. Надає можливість практичного використання отриманих знань базових дисциплін. 4. Користується попитом у роботодавців, підтримується інформаційно з боку компанії-розробника. 5. Виводить студента на початковий професійний рівень.
Чому можна навчитися (результати навчання)	<ol style="list-style-type: none"> 1. Основні поняття комп'ютерної криміналістики і правового забезпечення розслідувань інцидентів інформаційної безпеки. 2. Основи знань про алгоритми розслідувань інцидентів інформаційної безпеки, проведенні комп'ютерно-технічної експертизи. 3. Основні програмні засоби пошуку даних, що використовуються у Forensics.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ol style="list-style-type: none"> 1. Приймати рішення про те, які методи та інструменти використовувати під час проведення розслідування з комп'ютерної криміналістики. 2. Використовувати інструменти криміналістики даних (інструменти аналізу файлів, інструменти аналізу реєстру, інструменти аналізу Інтернет, інструменти аналізу електронної пошти, інструменти аналізу мобільних пристроїв, мережні інструменти комп'ютерної криміналістики тощо). 3. Використовувати інструменти OSINT (пошук інформації по відкритим джерелам). 4. Самостійно вивчати супутні технології комп'ютерної криміналістики.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, PCO, методичні рекомендації до виконання лабораторних робіт
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Бойко О.В.

Дисципліна	Бази даних безпілотних систем та автономної робототехніки
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання мови Python , знання HTML+CSS, , базові знання теорії баз даних, SQL, знання протоколів World Wide Web.
Що буде вивчатися	Архітектура програмного забезпечення дронів і мобільних роботів. Інформаційна модель навколишнього середовища мобільного робота. Структура інформаційного забезпечення мобільних роботів. Особливості використання в інформаційному забезпеченні мобільних роботів сучасних СУБД (реляційних PostgreSQL, MySQL, SQLite і NoSQL на прикладі MongoDB та ін.) . Використання ORM для спрощення обробки інформації на прикладі ORM Python Peewee та ORM SQLAlchemy. Реалізація доступу до даних СУБД та побудова інтерфейсу користувача інформаційної системи з

	допомогою фреймворків Flask, Django і концепції MVC . Поняття про ГІС (геоінформаційні системи) та їх використання для цілей навігації , локалізації, орієнтації в просторі та картографування. Збереження та конвертація інформації з використанням XML, JSON, CSV.
Чому це цікаво/треба вивчати	Python — сучасна універсальна мова програмування і є найпоширенішою в вбудованих системах управління в мобільній робототехніці і її можна використовувати для реалізації будь-якого класу задач від простого сценарію автоматизації до системного програмування, від рішення задач навігації та локалізації з використанням ГІС (геоінформаційної системи робота) до програмування задач управління мікроконтролерами та мікрокомп'ютерами дрона або робота, використання фреймворків Python для зв'язку з структурованою інформацією, яка потребує для своєї обробки різних типів БД (реляційних, NoSQL, графових, табличних і інших). Для обробки інформації і обміну з зовнішнім світом робот потребує розвинутої серверної частини, побудованої з використанням протоколів і стандартів WWW. Розробка серверної частини без фреймворку втомлює, тому фреймворки і закріпилися в якості основи веб-розробки мобільних роботів. Одним з найбільш поширених в світі є Python фреймворки Django , Flask .
Чому можна навчитися (результати навчання)	Розробляти інформаційне забезпечення мобільних роботів з використанням ви Python та СУБД різних типів та архітектури, використовувати для искорення розробки ORM; використовувати засоби автоматизації та фреймворки для розробки інформаційного забезпечення для безпілотних та автономних систем для серверної частини системи управління роботом;
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> • для розробки систем управління автономними мобільними роботами та дронами; • для вирішення задач навігації та локалізації для інших безпілотних систем.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, практичні та семінарські заняття
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Резніков С.А.

Дисципліна	Сучасні технології розробки WEB-застосунків на платформі Django (Python)
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: основ програмування, реляційні бази даних та основи SQL (можливе одночасне вивчення), основи HTML та CSS (можливе одночасне вивчення). Бажано основи JavaScript (не критично).
Що буде вивчатися	Способи аналізу функціональних вимог для сучасних Web-додатків; - створення проекту і Web-застосунків на основі Django; - розробка моделі об'єктно-реляційній проекції баз даних Django; - розробка системи Web-інтерфейсів користувача на основі Django, у тому числі з використанням Bootstrap та інших JavaScript фреймворків; - система тестування для Web-застосунків на основі Django; - мікроархітектура на основі Django Web-застосунків; - розгортання і контейнеризація Django Web-застосунків на Docker.

Чому це цікаво/треба вивчати	<p>За один семестр можна вивчити і отримати перший професійний досвід розробки Web-застосунків на основі Django.</p> <p>Цікаво для тих, хто маючи знання основ програмування, HTML та CSS бажає вийти на професійний рівень розробки Web-застосунків і планує після університету працювати в цій сфері, або почати свій бізнес.</p> <p>Тема курсу розрахована на опанування наступних елементів розробки Web-застосунків: розробка Web-інтерфейсів користувача, серверної частини Web-застосунків, об'єктно-реляційній проєкції баз даних, тестування Web-застосунків, бізнес-аналітики, документування, менеджменту, тощо.</p>
Чому можна навчитися (результати навчання)	<p>Проєктуванню і бізнес-аналітики Web-застосунку на основі Django.</p> <p>Повному циклу розробки Web-застосунку на основі Django.</p> <p>Розробці мікроархітектури на основі Django Web-застосунків і розгортанню програмної системи на основі контейнеризації.</p> <p>Професійному тестуванню Web-застосунку на основі Django.</p>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Отриманні знання, вміння, досвід і навички дають можливість працевлаштування у відповідній професійній сфері.</p> <p>Також отриманий досвід дозволяє зробити перший крок щодо створення власного бізнесу на основі Web-застосунків і програмних систем.</p>
Інформаційне забезпечення	<p>Силабус, методичні рекомендації до виконання лабораторних робіт, інформаційна підтримка у Телеграмі (група відкривається на час занять), середовище розробки (https://www.jetbrains.com/ru-ru/pycharm/), довідкова система (https://www.djangoproject.com/start/).</p>
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Катін П.Ю.

Дисципліна	Комп'ютерна графіка та обробка зображень
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання лінійної алгебри та аналітичної геометрії (вектори, матриці, векторні рівняння). Базові знання з інтегрального та диференційного числення. Основи програмування, базові знання ООП.
Що буде вивчатися	<p>Створення зображень з 3d моделей методом трасування променів (path tracing): рівняння рендерингу, перетин променів із тривимірними примітивами, моделі затінення, матеріали, BRDF, метод Монте-Карло.</p> <p>Realtime rendering: базова архітектура графічних процесорів, shader pipeline, представлення тривимірних моделей, растеризація, текстурування, матеріали, geometry/vertex/pixel shaders, PBR, пост-обробка.</p> <p>Обробка зображень: типові алгоритми перетворень, корекція кольору, tonemapping, формати/методи стискання зображень.</p>
Чому це цікаво/треба вивчати	<p>Цей курс має на меті познайомити слухача з тим, як працюють речі, із якими більшість із нас щоденно стикається. Прикладом таких речей можуть бути тривимірні графіка в сучасних іграх, комп'ютерні спецефектами у фільмах або просто зображення на моніторі. Навіть текст, який ви зараз читаете, надрукований шрифтом, що складається з кривих Безьє і є прикладом векторної графіки.</p> <p>Комп'ютерна графіка це область ІТ, в якій не просто оброблюються байти, передаються з сервера на сервер невидимі пакети, і не вщухають суперечки між прихильниками ООП та ФП. Натомість ви і будь-хто, навіть не розбираючись в темі, може побачити на власні очі результати роботи вашого</p>

	коду.
Чому можна навчитися (результати навчання)	Крім насамперед створення реалістичних зображень та обробки тривимірної графіки слухачі курсу матимуть змогу навчитися використовувати багатопоточність для розпаралелювання складних обчислювальних задач, писати доволі складні програми з монолітною архітектурою; познайомитися на практиці з ООП та SOLID та розширити коло знайомих алгоритмів (MC integration, spatial access methods, Huffman coding, LZW)
Як можна користуватися набутими знаннями і уміннями (компетентності)	Використовувати отримані знання для розробки комп'ютерних ігор або рушіїв комп'ютерних ігор.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	ст. викладач Ісаченко Г.

Дисципліна	Комп'ютерна графіка та мультимедіа
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	навички програмування на мові високого рівня; знання лінійної алгебри, геометрії та тригонометрії; розв'язування рівнянь, похідних, інтегралів; знання векторів, матриць; вміння розв'язувати системи рівнянь.
Що буде вивчатися	<ul style="list-style-type: none"> вступ до комп'ютерної графіки; програмування 2D графіки; трансформація та відображення об'єктів 3D графіки; моделювання 3D об'єктів та знаходження видимої поверхні; освітлення та рендеринг поверхні; алгоритми трасування променів та їх застосування.
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволяють розуміти принципи створення та редагування цифрових зображень, а також комп'ютерних ігор.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> пояснювати основні концепції комп'ютерної графіки, включаючи перегляд, проекцію, перспективу, моделювання та перетворення у двох та трьох вимірах; застосовувати концепції кольорних моделей, моделей освітлення та затінення, текстур, трасування променів, прихованої поверхні, згладжування та рендерингу; інтерпретувати математичні засади комп'ютерної графіки; описувати основи анімації, параметричні криві та поверхні, а також точкове освітлення; визначати типовий графічний конвеєр та застосувати методи графічного програмування для створення комп'ютерної графіки; створювати ефективні програми для вирішення задач програмування комп'ютерної графіки, включаючи 3D-трансформацію, моделювання об'єктів, кольорне моделювання, освітлення, текстури та трасування променів.
Як можна користуватися набутими знаннями і уміннями	<ul style="list-style-type: none"> створювати цифрові зображення та інтерактивну 3D графіку; розробляти комп'ютерні ігри та додатки.

(компетентності)	
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Родіонов П.Ю.

Дисципліна	Розробка ігрових застосувань (авторський курс компанії Genesis)
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	<ul style="list-style-type: none"> • базові знання з програмування • ООП • базові знання C# та розуміння синтаксису
Що буде вивчатися	<ul style="list-style-type: none"> • Побудова архітектури, • Нашарування динамічних, аудіо-візуальних ефектів, • Використання ШІ, мета-механіки. стек технологій: Unity + C#.
Чому це цікаво/треба вивчати	Незважаючи на обраний для курсу стек технологій, отриманні у ході виконання завдань знання стануть гарною базою не лише для розробки на Unity + C#, а і для поглиблення в спеціальність на інших технологіях на кшталт Unreal Engine чи Godot
Чому можна навчитися (результати навчання)	Повний курс створення та релізу мобільних ігор. Побудова архітектури, нашарування динамічних, аудіо-візуальних ефектів, використання ШІ, мета-механіки. стек технологій: Unity + C#.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Всі викладенні вище знання мають допомогти як у створенні власних ігрових проєктів, так і при потенціальному працевлаштуванні у ігрову сферу
Інформаційне забезпечення	Повний курс створення та релізу мобільних ігор
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	Авторський курс компанії Genesis

Дисципліна	Розподілені інформаційні системи
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку	Базові знання з дисциплін: Інженерія програмного забезпечення, Програмування, Об'єктно-орієнтоване

вивчення	програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Вступ до операційної системи Linux, Архітектура комп'ютерів, Дискретна математика
Що буде вивчатися	— архітектурні шаблони реалізації розподілених інформаційних систем, основи сервіс-орієнтованої архітектури, принципи та протоколи взаємодії веб-сервісів, принципи побудови програмних інтерфейсів, способи інтеграції розподілених програмних ресурсів; — основи програмної взаємодії з SQL- та NoSQL-базами даних, концепції потокового опрацювання даних; — основи технологічного оснащення процесу проектування, документування, розроблення та тестування програмного забезпечення; — основи побудови веб-застосунків (Single Page Application, SPA, Power Web Application, PWA) з використанням сучасних програмних засобів (React, Angular, Vue); — способи реалізації процесу збирання та розміщення програмних систем з використанням хмарних сервісів PAAS.
Чому це цікаво/треба вивчати	— сьогодні володіння знаннями, вміннями і навичками, пов'язаними з процесом проектування розподілених систем колективного використання, процесом розроблення відповідного програмного забезпечення та організації технологічної підтримки всіх стадій життєвого циклу програмного забезпечення, є однією з ключових конкурентних переваг на ринку праці в сфері ІТ.
Чому можна навчитися (результати навчання)	— виконувати проектування інформаційних систем на основі використання сучасних архітектурних принципів із застосуванням засобів моделювання та документування на всіх стадіях життєвого циклу програмного забезпечення, розробляти інформаційне та програмне забезпечення таких систем та організовувати технологічну підтримку процесу їх розроблення та розміщення.
Як можна користуватися набутими знаннями і уміннями (компетентності)	— виконувати проектування та розроблення сучасних інформаційних систем колективного використання, організовувати технологічну підтримку процесу розроблення програмного забезпечення, використовувати засоби для організації розподіленого оброблення даних, використовувати реляційні, документо-орієнтовані та графові бази даних, інтегрувати власне програмне забезпечення з загально доступними веб-сервісами, здійснювати розміщення програмного забезпечення в хмарному середовищі PAAS.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Болдак А. О.

Дисципліна	Паралельні та розподілені обчислення
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Архітектура комп'ютерів, Аналітична геометрія, Інженерія програмного забезпечення
Що буде вивчатися	Методи організації обчислень в паралельних комп'ютерних системах з різною структурною організацією: систем зі спільною пам'яттю, систем з

	<p>локальної пам'яттю, розподілених систем. Життєвий цикл розробки програми для паралельної системи, що включає:</p> <ul style="list-style-type: none"> - розробку паралельного алгоритму обчислень . - розробку алгоритмів виконання потоків з рішенням завдання взаємного виключення та синхронізації. - розробку структурної схеми взаємодії потоків для обраних мов (бібліотек) паралельного програмування та засобів організації потоків: семафорів, мютексів, подій, критичних секцій, - методи налагодження і тестування паралельної програми - дослідження відсотку завантаження процесорів (ядер) системи, визначення часу виконання, обчислення коефіцієнту прискорення (speedup). <p>В лабораторних роботах використовуються сучасні мови та бібліотеки паралельного програмування: Java, C#, WinAPI, Ada, OpenMP, MPI, TBB.</p>
Чому це цікаво/треба вивчати	Сучасні комп'ютери (починаючи від планшету і закінчуючи супер комп'ютерами) обладнано багатоядерними процесорами, тому створення ефективного програмного забезпечення для них потребує використання потоків для оптимального використання багатоядерного обладнання. Багатоядерні процесори також є частиною більшості гаджетів, таких як смартфони, mp3 плеєри, відеоплеєри, ігрові приставки, що дозволяє прискорити (покращити) обробку звуку та зображення.
Чому можна навчитися (результати навчання)	Вмінню розробляти програмне забезпечення для паралельних та розподілених комп'ютерних систем різного призначення, аналізувати завдання на наявність паралелізму, будувати оптимальні паралельні алгоритми, використовувати потоки для побудови програми, організувати оптимальну взаємодію потоків в залежності від структури комп'ютерної системи, виконувати налагодження паралельної програми, запобігати та позбавлятися тупикових ситуацій. Використанню сучасних мов та бібліотек паралельного програмування.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані базові знання допоможуть у подальшому професійному навчанні в тому числі самостійно, розширюють перспективу кар'єрних можливостей в галузі розроблення програмного забезпечення для будь яких паралельних комп'ютерних систем, систем реального часу, а також створення серверно-клієнтських додатків.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, практичні та семінарські заняття, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Корочкін О.В.

Дисципліна	Гібридні комп'ютерні системи
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Теорія ймовірності та математична статистика, Системне програмування, Системне програмне забезпечення, Комп'ютерні системи, Архітектура комп'ютерів, Комп'ютерні мережі
Що буде вивчатися	Основи проектування аналогових та гібридних операційних блоків, аналогово-цифрових та цифро-аналогових перетворювачів інформації,

	аналогових та гібридних процесорів, методи математичного моделювання з допомогою операційних блоків, методи підготовки задач для аналогових та гібридних систем, методи програмування для аналогових та гібридних систем для спеціалістів фахового напрямку "Комп'ютерна інженерія", забезпечити достатню глибину засвоєння дисциплін основної інженерної підготовки зі спеціальності, а також для оволодіння сучасними технічними засобами та способами програмування аналогових та гібридних систем для наступних спеціалізації.
Чому це цікаво/треба вивчати	Отримані знання дозволять оволодіти основними методами побудови нетрадиційних комп'ютерних систем
Чому можна навчитися (результати навчання)	Проектуванню систем з гетерогенною структурою, наприклад систем з CPU і GPU
Як можна користуватися набутими знаннями і уміннями (компетентності)	Після проходження курсу студенти зможуть проектувати високопродуктивні комп'ютерні системи
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Селіванов В.Л.

Освітній компонент з 8Ф-каталогу

Дисципліна	Технології та протоколи мультисервісних мереж
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: математика, фізика, комп'ютерні мережі, інтернет-технології
Що буде вивчатися	Використання та створення інформаційно-комунікаційних технологій, використання алгоритмічних мов та середовищ розробки у процесах проектування, розробки, розповсюдження та використання мультисервісних мереж. Особливості і порядок застосування базових технологій розробки мережевих протоколів на базі стандартів еталонних моделей OSI/ISO та стеків протоколів TCP/IP, IPX/SPX, SS7 (рівні MTP, SCCP, MAP, ISUP), кореляція SS7 з моделлю OSI; знання, розуміння та застосування VoIP/SIP: загальні принципи, callflow; кореляція стеку протоколів TCP/IP з моделлю OSI (призначення протоколів, адресація, маршрутизація, загальні принципи функціонування IP-мереж). Застосування протоколів маршрутизації OSPF, RIP, BGP, EIGRP в мультисервісних мережах. Особливості застосування технологій PDH, SDH, xWDM, MPLS. Протоколи IP, HTTP та DHCP для WEB – технологій. DHCPv4 та DHCPv6. Особливості застосування SMPP. Поняття мобільності в стільникових мережах зв'язку. Структури GSM мережі (включно з призначенням та функціями її елементів - HLR, MSC, VLR, SMSC, GMSC, SGSN, GGSN тощо). Структура LTE мережі (архітектура та призначення основних елементів).
Чому це цікаво/треба вивчати	Стрімкий розвиток сучасних інформаційних технологій надає нові можливості для використання на різних рівнях комп'ютерних мереж, мережевих протоколів різної функціональності для розвитку відповідних

	інформаційних сервісів і організації взаємодії інформаційного простору з їх партнерами та клієнтами, використання мобільного середовища та різномірних джерел інформації і пристроїв комунікації.
Чому можна навчитися (результати навчання)	Знання та розуміння моделі OSI/ISO. Знання, розуміння та застосування стеку протоколів SS7. Знання, розуміння та застосування VoIP/SIP. Знання, розуміння та застосування стеку протоколів TCP/IP. Знання, розуміння та застосування PDH, SDH, xWDM, MPLS. Знання, розуміння та застосування SMPP та SIGTRAN (основні рівні, що відповідають за передачу трафіку та їх основні мітки). Знання та розуміння поняття мобільності в стільникових мережах зв'язку та яким чином вона забезпечується. Знання та розуміння структури GSM та LTE мереж.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші). Здатність управляти якістю продуктів і сервісів інформаційних систем та технологій протягом їх життєвого циклу. Здатність вибору, проектування, розгортання, інтегрування, управління, адміністрування та супроводжування інформаційних систем, технологій та інфокомунікацій, сервісів та інфраструктури організації. Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, електронний конспект та презентація
Форма проведення занять	Лекції, комп'ютерний практикум
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	д.т.н., професор Жураковський Б.Ю.

Дисципліна	Програмні засоби проектування та реалізації нейромережових систем
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: Інтелектуальні системи та технології, програмування мовою Python
Що буде вивчатися	Проектування, реалізація та дослідження програмних систем з використанням штучного інтелекту. Методи створення, навчання і функціонування нейромережових систем та систем глибокого навчання. Програмні засоби реалізації та дослідження нейромережових систем на Python з використанням бібліотек TensorFlow і Keras.
Чому це цікаво/треба вивчати	Машинне навчання – розділ інформатики, який вивчає розробку алгоритмів, які можна навчати. Одним з розділів машинного навчання

	<p>є штучні нейронні мережі. Глибоке навчання є однією з найгарячіших галузей науки про дані, оскільки багато тематичних досліджень дають вражаючі результати в робототехніці, розпізнаванні зображень та штучному інтелекті.</p> <p>Однією з найпотужніших і простих у використанні бібліотек Python для розробки та оцінки програмних моделей глибокого навчання є Keras. Він обгортає ефективні чисельні бібліотеки обчислень Theano та TensorFlow. Вивчення даних програмних технологій реалізації та дослідження нейромережевих систем дозволить легко та просто реалізовувати складні програмні системи штучного інтелекту.</p>
Чому можна навчитися (результати навчання)	<p>При вивченні даної дисципліни, студенти вивчать теоретичні основи нейронних мереж та отримують початковий досвід в розробці програмного забезпечення, що реалізує нейромережеві технології. На лабораторних заняттях отримують початковий досвід створення програмних систем на основі нейронних мереж. Проведуть розробку та дослідження нейромережевих систем різного призначення за допомогою мови програмування Python та бібліотек TensorFlow і Keras.</p>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Набуті знання можна використати при розробці програмних систем зі штучним інтелектом.</p>
Інформаційне забезпечення	<p>Силабус, конспект лекцій, презентаційні матеріали, методичні вказівки до виконання лабораторних робіт</p>
Форма проведення занять	<p>Лекції, лабораторні роботи</p>
Семестровий контроль	<p>Залік, 3 курс 6 семестр</p>
Викладач	<p>к.т.н., доцент Шимкович В.М.</p>

Дисципліна	Професійне використання SQL та PL/SQL на прикладі РСУБД Oracle 11g
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	<p>Базові знання з теорії реляційних баз даних та проектування реляційних баз даних в розділі інфологічного та фізичного моделювання. Основи роботи в SQL, мінімальне адміністрування та налаштування серверу Oracle Database 11g XE.</p>
Що буде вивчатися	<p>В курсі буде вивчатись PL/SQL в розділах створення збережуваних процедур, функцій, тригерів, динамічний SQL та PL/SQL. Більш глибоко відпрацюємо запити на виборку, складні запити, аналіз та оптимізація запитів.</p>
Чому це цікаво/треба вивчати	<p>При створенні програмних застосунків важливе місце відводиться саме бази даних, написанню скриптів та запитів для коректної, швидкої роботи застосунків. PL/SQL одна з найуживаніших мов на рівні T-SQL, PL/pgSQL та інші. Знання базових конструкцій підвищить конкурентоспроможність розробника серед інших претендентів.</p>
Чому можна навчитися (результати навчання)	<p>Ознайомитись і навчитись писати складні конструкції мови PL/SQL, аналізувати запити DML, застосовувати PL/SQL для певного програмного застосування.</p>
Як можна	<p>Набуті знання можна застосовувати при проектуванні WEB застосунків,</p>

користуватися набутими знаннями і уміннями (компетентності)	адмініструванні реляційних серверів баз даних, програмування мовою PL/SQL.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, методичні вказівки з виконання лабораторних робіт (планується), конспект лекцій (планується), презентаційний матеріал та файли з прикладом скриптів (планується).
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Ульяницька К.О.

Дисципліна	Розробка мікросервісів
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Технології розробки веб-застосунків.
Що буде вивчатися	<ul style="list-style-type: none"> • Microservices introduction • Discovery Server • API Gateway • Centralized Configuration • Security • Container Management • Cloud Hosting • Scaling up • Operational Readiness (centralized logging, metrics) • Tracing • Message Broker Web sockets
Чому це цікаво/треба вивчати	Розробка мікро-сервісів для використання в хмарах є напрямом, який є затребуваним роботодавцем. Курс містить важливі аспекти розробки, розгортання та моніторингу мікро-сервісів.
Чому можна навчитися (результати навчання)	Розробка, розгортання, моніторинг мікро-сервісів.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання та навички можна використати при працевлаштуванні.
Інформаційне забезпечення	Навчальна та робоча програма дисципліни, РСО, практичні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Сирота О.П.

Дисципліна	Основи PHP (авторський курс компанії Genesis)
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин

Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	– базові знання з програмування – розуміння ООП робота з базами даних
Що буде вивчатися	мова PHP
Чому це цікаво/треба вивчати	PHP спеціально сконструйований для веб-розробок і його код може впроваджуватися безпосередньо в HTML.
Чому можна навчитися (результати навчання)	Повний курс освоєння мови програмування загального призначення із відкритим вихідним кодом
Як можна користуватися набутими знаннями і уміннями (компетентності)	Веб-розробка
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, документація
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	Авторський курс компанії Genesis

Дисципліна	Розподілені системи та обчислення
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання операційних систем або системного програмування, об'єктно-орієнтованого програмування та комп'ютерних мереж
Що буде вивчатися	<ul style="list-style-type: none"> • Основні поняття розподілених систем • Основи розробки розподілених програмних систем, підходи до їх реалізації (Sockets, Remote Procedure Call, REST, Message Brokers) та формати передачі даних (XML, JSON) Засоби мов програмування для розробки програмного забезпечення розподілених систем
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> • Отримані знання дозволяють використовувати можливості сучасних комп'ютерів для створення розподілених програмних чистем систем; принципи роботи розподілених програмних систем та обирати засоби взаємодії між компонентами розподілених програмних систем
Чому можна навчитися (результати навчання)	аналізувати можливості по пришвидшенню роботи або підвищенню надійності програмного забезпечення за рахунок створення розподіленого програмного забезпечення; організувати взаємодію розподілених програмних систем
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> • Створювати розподілені програмні системи • Використовувати бібліотеки мов програмування для ефективної передачі даних між програмними компонентами Обирати засоби та формати даних для розподіленої взаємодії
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, контрольні завдання, матеріали для лекцій
Форма проведення	Лекції та лабораторні (комп'ютерні практикуми)

занять	
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	

Дисципліна	Цифрова обробка сигналів
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська, англійська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Вища математика, Програмування, Дискретна математика, Архітектура комп'ютерів, Комп'ютерні системи
Що буде вивчатися	Математичні основи та алгоритми цифрової обробки сигналів (ЦОС) і зображень, а також методи їх складання, програмування та апаратної реалізації.
Чому це цікаво/треба вивчати	Сучасні апаратні та програмні засоби, що поширені в ІТ-галузі дуже часто ґрунтуються на виконанні алгоритмів ЦОС. Це, наприклад, засоби автоматизації, телекомунікації, обробки зображень, комп'ютерні ігри, розпізнавання образів та штучний інтелект, IoT. Без знань алгоритмів ЦОС практично неможливо тестувати, модернізувати та розробляти такі засоби.
Чому можна навчитися (результати навчання)	— вибирати, моделювати та модернізувати алгоритми ЦОС; — розробляти застосунки, які виконують алгоритми ЦОС.
Як можна користуватися набутими знаннями і уміннями (компетентності)	— проводити аналіз, діагностування, вибір існуючих засобів ЦОС, — виконувати модернізацію існуючих та розробку нових засобів ЦОС, — впроваджувати алгоритми ЦОС у засобах IoT та штучного інтелекту. Знати основи ЦОС та мати досвід її програмування — це обов'язкова вимога при прийомі на роботу у більшості фірм, що займаються розробкою та поширенням застосунків та апаратури для автоматизації, телекомунікації, штучного інтелекту, IoT.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	д.т.н., професор Сергієнко А.М.

Дисципліна	Математичні основи захисту даних та інформаційної безпеки
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з архітектури комп'ютера, комп'ютерних систем та комп'ютерних мереж, алгоритмів та структур даних, дискретної математики, теорії ймовірності та математичної статистики, системного програмування.
Що буде вивчатися	Математичні основи сучасної криптографії: незворотні перетворення булевої алгебри, теорії чисел, еліптичних кривих та кінцевих полів Галуа.

	Методи синтезу криптографічних булевих перетворень, побудова на їх основі алгоритмів захисту даних. Сучасні алгоритми та протоколи захисту інформації на основі булевих функцій. Атаки на алгоритми та протоколи цього класу. Методи побудови систем криптографічного захисту на основі незворотних перетворень теорії чисел. Алгоритми та протоколи захисту інформації на основі незворотних перетворень теорії чисел та методи їх зламу. Математичні основи організації гомоморфного шифрування даних при їх обробці в хмарах. Методи зламу засобів захисту даних шляхом часового чи статичного аналізу динаміки споживання потужності. Математичні засади програмного поліморфізму при захищеній організації обчислень. Квантові обчислення та квантова криптографія.
Чому це цікаво/треба вивчати	Поступальний розвиток інформаційної інтеграції вимагатиме якісно нових засобів захисту даних та розмежування доступу до них в інтегрованих середовищах. Створення та використання таких засобів неможливе без оволодіння фундаментальними основами криптографії. Професійний аналіз реального рівня захищеності даних та програм вимагає наявності спеціальної математичної підготовки.
Чому можна навчитися (результати навчання)	Отримати ґрунтовні знання в спеціальних розділах математики, які лежать в основі сучасних та перспективних засобах криптографічного захисту даних. Оволодіти навичками створення базових блоків криптографічного захисту даних, побудови на їх основі алгоритмів та протоколів. Набути вмінь аналізувати рівень захищеності засобів інформаційної безпеки. Знати протоколи криптографічного захисту інформації в комп'ютерних системах та мережах.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробляти, вдосконалювати та адаптувати під умови конкретного використання програмні і апаратні засоби криптографічного захисту даних. Створювати гомоморфні шифри для захисту даних безпосередньо в процесі їх віддаленої обробки в хмарах залежно від її характеристик. Здійснювати професійний аналіз інформаційної безпеки даних з огляду на здатність протистояння різним видам атак.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	к.т.н., доцент Марковський О. П.

Дисципліна	Технології Computer Vision
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Алгоритми та методи обчислень, Системне програмування, Системне програмне забезпечення, Алгоритми і структури даних, Організація баз даних, Програмування, Архітектура комп'ютерів, Комп'ютерні мережі, Дискретна математика, Комп'ютерна логіка, Інженерія програмного забезпечення, Паралельне програмування
Що буде вивчатися	Дисципліна має на меті: надання ґрунтового теоретичного базису та потужних практичних навичок програмної реалізації методів, математичних моделей і алгоритмів технологій Computer Vision (комп'ютерного бачення = комп'ютерного зору). Теоретичні основи Computer Vision надаються у формі лекційних

	<p>занять з обов'язковою демонстрацією практичної реалізації розглянутих алгоритмів у формі прикладів програмного коду.</p> <p>Практичні навички застосування технологій Computer Vision набуваються на лабораторних заняттях, які побудовані за принципом нарощування функціональності розроблених скриптів. При цьому особлива увага приділяється процесам інженерії програмного забезпечення.</p> <p>Практична частина дисципліни орієнтована на застосування мови програмування високого рівня Python з вивченням можливостей графічних бібліотек Graphics, Tkinter, Matplotlib, NumPy (для «сирої» реалізації алгоритмів Computer Vision) та спеціалізованих пакетів типу PIL/ Pillow, OpenGL, OpenCV для створення програмних модулів завершеної практичної спрямованості.</p> <p>Дисципліна розкриває суть етапів класичного конвеєра обробки цифрових графічних зображень: синтез зображення (просторове перетворення) – растеризація (реалістичне зображення, цифрова обробка) – векторизація (ідентифікація об'єкту та цільова обробка зображень). Зазначене реалізується у таких темах:</p> <ol style="list-style-type: none"> 1. Синтез цифрового зображення (2D, 3D – об'єкти, графічний метод, аналітичні та неаналітичні методи); 2. Базові геометричні перетворення 2D, 3D об'єктів (масштабування, переміщення, обертання, проекції); 3. Растеризація і растрові зображення, принципи створення, характеристики та алгоритми обробки (растеризація, зміна яскравості, фільтрація); 4. Векторизація і векторні зображення, математичні моделі, алгоритми апроксимації, інтерполяції та згладжування (МНК, Сплайни, Криві Безьє тощо); 5. Моделі реалістичних зображень, доповнена реальність, моделі світла і тіні: 3D-конвеєр, рендерінг; 6. Цифрова обробка зображень: фільтрація; корекція кольору; сегментація; векторизація; морфологічна обробка; розпізнавання. 7. Методологічні, технологічні та прикладні аспекти Computer Vision: обробка сенсорних даних; навігація; штучний інтелект; управління.
<p>Чому це цікаво/треба вивчати</p>	<p>Опанування знань, умінь та навичок, що потребують посади: Software Developer with Computer Vision; Embedded developer for Computer Vision systems; Computer Vision Research Engineer тощо.</p>
<p>Чому можна навчитися (результати навчання)</p>	<ol style="list-style-type: none"> 1. Знання та навички застосування і реалізації базових алгоритмів Computer Vision: синтез 2D, 3D графічних об'єктів та їх геометричні перетворення; синтезу та обробки растрових зображень; синтезу та обробки векторних зображень; формування реалістичних зображень; цифрової обробки зображень; прикладних аспектів Computer Vision; комплексної реалізації етапів конвеєру цифрової обробки зображень для задач Computer Vision та інших прикладних галузей. 2. Знання і навички застосування базових бібліотек Python для реалізації алгоритмів Computer Vision: Matplotlib, NumPy PIL/ Pillow, OpenCV.
<p>Як можна користуватися набутими знаннями і</p>	<p>Набуті компетенції з Computer Vision можливо застосовувати на проектах наступних прикладних галузей і технологій:</p>

уміннями (компетентності)	1. Реконструкція 3D сцени та навігація в ній (Structure-from-Motion, Road Scene Understanding and Autonomous Driving); 2. Сегментація сцени і розуміння сюжету (Significance maps, video and 3D segmentation, 3D stream, multiple tracking, object detection, activity detection and prediction, group analysis, object detection and recognition); 3. Розпізнавання зображень, ідентифікація об'єктів (Convolution networks, Deep learning, Image Retrieval, Object Detection).
Інформаційне забезпечення	Навчальна і робоча програма дисципліни, PCO, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік, 3 курс 6 семестр
Викладач	д.т.н., професор Писарчук О.О.

4 курс 7 семестр

Освітній компонент з 9Ф-каталогу

Дисципліна	Основи WEB – технологій
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: знання мов програмування, базові знання роботи з базами даних, робота в мережі інтернет.
Що буде вивчатися	Клієнтська частина: VSCode, GitHUB, HTML5, CSS3, JavaScript, Об'єкти в JavaScript. Об'єктна модель документа (DOM - document object model). Події в JavaScript. Асинхронність в JavaScript Серверна частина: Розробка серверних додатків мовою Node.js . Робота з мережними запитами. Аутентифікація та авторизація. REST API, WebSockets. Socket.IO. Розробка серверних додатків мовою PHP . Робота з базою даних MySQL.
Чому це цікаво/треба вивчати	Fullstack-розробники – універсальні програмісти, які розробляють і серверну, і клієнтську частину проекту. Ці фахівці дуже затребувані, тому що можуть замінити програмістів різного профілю. Fullstack-розробники володіють широким стеком технологій, тому можуть створити веб-проект поодиночі.
Чому можна навчитися (результати навчання)	Розробляти сайти статичної та динамічної структури. Розробляти інтерфейс користувача для веб-додатків. Розробляти структуру бази даних для веб-додатків. Створювати серверні програми, які обробляють інформацію, що зберігається у БД.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> ▪ для розробки статичних сторінок; ▪ для розробки адаптивних WEB-сторінок; ▪ для розробки динамічних WEB-сторінок ▪ для розробки інтерфейсу користувача за допомогою JS. ▪ розробляти серверні додатки мовою Node.js; ▪ розробляти серверні додатки мовою PHP; розробляти структуру БД та забезпечувати підключення веб-додатків до бази даних;
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс, VSCode, GitHUB, HTML5, CSS3, Node.js, PHP, MySQL
Форма проведення	Лекції, лабораторні роботи

занять	
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	к.т.н., доцент Голубев Л.П.

Дисципліна	Безпроводні технології для управління смарт-середовищами
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: математика, фізика, системотехніка, комп'ютерні мережі
Що буде вивчатися	Екосистема Інтернету речей. Архітектура Інтернету Речей (Рівень датчиків, Мережевий рівень, Рівень обробки даних, Рівень додатків). Еталонні моделі IoT. IoT платформи (платформа Linux Foundation, платформа AggreGate, платформа Everyware Cloud). Моделі IoT шлюзів. Прості та інтелектуальні сенсори. Сенсорно-комп'ютерні системи. Безпроводна передача та обробка даних Інтернет речей. Передача даних від датчиків до хмарних сховищ. Обробка даних отриманих за допомогою датчиків. Технології та протоколи передачі даних на довгі відстані в IoT мережах: LoRaWAN, SigFox, NB-IoT, Weightless-P. Технології та протоколи передачі даних на короткі відстані в IoT мережах: Z-Wave, NFC, RFID, Bluetooth Low Energy, Wi-Fi HaLow. Сенсорні мережі. Смарт грид на рівні квартири або будинку. Системи на базі технологічної платформи Смарт грид. Smart City. Платформи Smart City. Smart Home. Smart Factory. Smart Manufacturing.
Чому це цікаво/треба вивчати	Смарт-середовище - це місце, в якому традиційні системи працюють більш ефективно за рахунок використання інформаційно-комунікаційних технологій. Інформаційно-комунікаційні технології дозволяють використовувати менше енергетичних ресурсів, задовольняючи незмінний обсяг потреб, та зменшувати масштаби парникової емісії. Це означає запровадження «розумнішої» системи міського транспорту, оновленої системи водопостачання та утилізації відходів, а також створення ефективніших систем опалення та охолодження будинків тощо.
Чому можна навчитися (результати навчання)	Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності. Розуміти знання технології, принципів організації та функціонування IoT, вміти проектувати IoT-системи, демонструвати знання мікроконтролерів, платформ налаштування, програмування кінцевих пристроїв, створювати вбудоване та серверне програмне забезпечення. Використовувати моделі та методи розподілу функціоналу між пристроями та взаємодії пристроя та середовища. Уміти перетворювати звичайні технічні вироби в інтернет речі, реалізовувати взаємодію інтернет речей між собою та середовищем.
Як можна користуватися набутими знаннями і умінями (компетентності)	Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, електронний конспект та презентація

Форма проведення занять	Лекції, практичні заняття
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	д.т.н., професор Жураковський Б.Ю.

Дисципліна	Розробка ігрових застосувань. Unity рішення
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: основ програмування і алгоритмізації. Бажані знання однієї з мов програмування C#, або C++, або JavaScript. Знання дискретної математики і основ аналітичної геометрії (не критично, вивчається у процесі).
Що буде вивчатися	Способи аналізу функціональних вимог, дизайну і сценарію ігрових програм на основі сучасних актуальних рушіїв; - термінологія і загальні аспекти створення комп'ютерних ігор; - види сучасних ігрових рушіїв для створення 2D, 3D комп'ютерних ігор; - досконале вивчення одного з обраних ігрових рушіїв (у першому наближенні Unity), для студентів з досвідом дозволено самостійно обирати тип ігрових рушіїв і вивчати його поглиблено; - розробка дизайну гри, комп'ютерної графіки; - розробка програмування (скриптів гри), типові патерни (шаблони) програмування, основи програмної архітектури ігрового рушія; - основи штучного інтелекту і математична підтримка розробки; - основи бізнесу ігрової комп'ютерної індустрії.
Чому це цікаво/треба вивчати	За один семестр можна вивчити базові основи розробки комп'ютерних ігрових програм. Отримати перший досвід розробки ігрових програм, визначитись з потенційним напрямком роботи у цій сфері: - дизайнер, художник ігрових програм; - розробник ігрових програм, серверна частина, ігрова механіка; - бізнес аналітик, тестер, та інші споріднені професійні напрямки. Визначитись з питаннями щодо стартапу або початку власного бізнесу.
Чому можна навчитися (результати навчання)	Проектуванню і бізнес-аналітики ігрових програм на основі сучасних актуальних рушіїв (на прикладі Unity), можливий інший ігровий рушії. Повному циклу розробки ігрових програм. Самостійному аналізу ігрової індустрії і набуттю знань з питань розробки і проектуванню комп'ютерних ігор.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отриманні знання, вміння, досвід і навички дають можливість працевлаштування у відповідній професійній сфері. Також отриманий досвід дозволяє зробити підґрунтя щодо створення власного бізнесу у галузі комп'ютерних ігор.
Інформаційне забезпечення	Силабус, методичні рекомендації до виконання лабораторних робіт, інформаційна підтримка у Телеграмі (група відкривається на час занять), середовище розробки, довідкова система (https://unity.com).
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	к.т.н, доцент Катін П.Ю.

Дисципліна	Розробка ігрових застосувань
-------------------	-------------------------------------

Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базова дискретна математика Теми, що стосуються програмування потребують знань з ООП.
Що буде вивчатися	<ol style="list-style-type: none"> 1. Комп'ютерна графіка 2. Штучний інтелект для ігор 3. Технічні теми: програмування (проектування ігор, паттерни, мережева взаємодія) <p>Повний перелік тем можна побачити на відео-лекціях за посиланням: https://www.youtube.com/playlist?list=PLkgXLMuasx7C7yMUsaq366htPg9rpM2lw</p>
Чому це цікаво/треба вивчати	Курс дозволяє за один семестр комплексно ознайомитись з тематикою ігрової розробки, для тих кому це цікаво або планує після університету працювати в цій сфері. Теми курсу розраховані на огляд різних аспектів розробки ігор (графіка, AI, геймдизайн, проектування), для того щоб а) розібратися з і більш цікавими для себе темами і б) мати загальне розуміння термінології, задач що виникають в ігровій розробці та методів їх вирішення
Чому можна навчитися (результати навчання)	<ol style="list-style-type: none"> 1. Проектуванню штучного інтелекту для різних видів ігор 2. Підходам до проектування гри як до великого програмного забезпечення <p>Загальному розумінню сучасного стану комп'ютерної графіки</p>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Всі викладенні вище знання мають допомогти як у створенні власних ігрових проєктів, так і при потенціальному працевлаштуванні у ігрову сферу
Інформаційне забезпечення	Всі лекції викладаються на ютуб каналі): https://www.youtube.com/channel/UC2Wuo5-6CauJ_jgigV-6JDQ Анонси можна читати в телеграмі: @gamedevkpi Обговорення в чаті: @kpigamedev chat
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	ст. викладач Коротенко А.

Дисципліна	Ігровий штучний інтелект
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	знання комп'ютерної дискретної математики, алгоритмів та структур даних, основ проектування алгоритмів, підходів до програмування, (наприклад, об'єктно-орієнтоване програмування), засобів візуалізації інтерфейсу користувача
Що буде вивчатися	<ul style="list-style-type: none"> • інтелектуальні агенти та середовища; • інтелектуальний пошук та взаємодія; • управління інтелектуальними агентами; • моделювання поведінки інтелектуальних агентів; • прийняття простих та складних рішень інтелектуальними агентами; <p>застосування нейронних мереж в ігровому штучному інтелекті.</p>

Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволять виробити певне мислення до процесу розробки ігрового штучного інтелекту та його складових
Чому можна навчитися (результати навчання)	вивчити комплексний підхід до проектування та програмування штучного інтелекту у комп'ютерних іграх
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> • програмування поведінки ігрового середовища в цілому та програмування інтелектуальних агентів (NPC, ботів, мобів) у ігровому середовищі; • застосування алгоритмічних рішень для розробки ігрового штучного інтелекту різних рівнів складності; використання нейронних мереж для адаптації комп'ютерного опонента під стратегії гравця у комп'ютерних іграх.
Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	ст. викладач Головченко М.М.

Дисципліна	Створення інформаційних систем (ІС) на базі Low-code платформ
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Володіння навиками бізнес-моделювання Володіння навиками програмування
Що буде вивчатися	<ul style="list-style-type: none"> • BPMN - моделювання • Технологія Low-Code систем Практичні аспекти використання
Чому це цікаво/треба вивчати	Low-code — це підхід до створення, налаштування і модифікації ІС, який практично не вимагає написання програмного коду. Low-code платформи використовують візуальні інтерфейси та діаграми представлення бізнес-процесів (наприклад, у BPMN-нотації). Це дає можливість використовувати результати роботи аналітиків напряму (автоматично) без участі розробників в процесі побудови ІС. Технологія потребує написання програмного коду в мінімальному об'ємі.
Чому можна навчитися (результати навчання)	Теоретично та практично розібратись з технологіями створення ІС с написанням програмного коду в незначних об'ємах. Створювати ІС на базі застосування Low-code систем
Як можна користуватися набутими знаннями і уміннями (компетентності)	Створювати інформаційні системи, користуючись даними проведення бізнес-аналізв
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, документація (в тому числі онлайн) на системи класу Low-code
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	

Дисципліна	
-------------------	--

	Мобільні комп'ютерні мережі
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Комп'ютерні мережі, Системне програмування, Архітектура комп'ютерів
Що буде вивчатися	Основні концепції побудови, архітектура, комунікаційна система мобільних комп'ютерних мереж, бездротові широкосмугові мережі, бездротової персональної мережі, бездротові 5G мережі, програмно-конфігуровані мобільні мережі, мобільні хмарні технології, мобільний інтернет, основи безпеки мобільних комп'ютерних мереж,
Чому це цікаво/треба вивчати	Курс спрямований на формування теоретичних знань та практичних навичок із побудови, керування, модернізації, моніторингу сучасних мобільних комп'ютерних мереж.
Чому можна навчитися (результати навчання)	Засвоїти принципи формування теоретичних знань та практичних навичок із побудови, експлуатації, моніторингу та аналізу продуктивності сучасних мобільних комп'ютерних мереж.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здатність підібрати тип, планувати і реалізовувати мобільні комп'ютерні мережі, керувати мережними ресурсами, підібрати комплекс необхідних апаратно-програмних засобів для мобільної комп'ютерної мережі, розширювати і модернізувати мережі, здійснювати моніторинг та аналіз продуктивності, діагностувати та розв'язувати проблеми.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	д.т.н., професор Кулаков Ю.О.

Дисципліна	Основи комп'ютерного моделювання
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Вступ до операційної системи Linux, Дискретна математика, Інженерія програмного забезпечення, Теорія ймовірності
Що буде вивчатися	- Імітаційні, математичні, логічні, еволюційні, структурно-лінгвістичні методи моделювання. Основи аналогового та фізичного моделювання. Методологія моделювання, як технології наукових досліджень та проектування комп'ютерних систем та їх програмного забезпечення. - Методи створення моделей з урахуванням вимог до їх адекватності, способи та прийоми їх програмної реалізації та використання на практиці, методики оцінки отриманих за допомогою моделей результатів. Використання моделей для оптимізації розробки та оцінки її надійності. Методи оцінки адекватності моделей. - Найбільш поширені на практиці моделі комп'ютерних систем – мережі Петрі та програмні пакети моделювання. - Математичні моделі оптимізації процесів та прийоми їх програмної реалізації. Динамічне та цілочисельне

	програмування. Методи еволюційного моделювання на основі генетичних алгоритмів та технологій штучного інтелекту.
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> - отриманні знання та практичні навички дозволяють науково та методично грамотно застосувати моделювання для вирішення широкого спектру практичних задач проектування систем, програмного забезпечення, організації передачі даних в локальних та глобальних мережах, оцінки надійності програм та рівня інформаційної безпеки. - віднаходити за допомогою моделювання оптимальні рішення проектних рішень та природничо-наукових та економічних проблем. - проводити аналіз поведінки систем та програм в різних, в тому числі критичних ситуаціях, а також при наявності помилок при їх розробці.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - Аналізувати можливості вирішення широкого кола практичних задач за допомогою методів моделювання; здійснювати обґрунтований вибір моделі та її побудову з урахуванням вимог до її адекватності, грамотно оцінювати надійність отримані результатів моделювання; - оволодіти практичними способами та прийомами програмування імітаційних, математичних та еволюційних моделей - Використовувати математичні моделі оптимізації процесів та динамічного програмування для створення ефективних програм. - Оцінювати з використанням комп'ютерних моделей параметри якості систем та програм, зокрема надійності їх роботи в різних ситуаціях, а також рівень безпеки даних та програм
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> - Використовувати арсенал методів комп'ютерного моделювання в якості інструменту наукових досліджень в технічних, природничо-наукових, медичнських, економічних та соціальних сферах - Застосовувати методи комп'ютерного моделювання та існуючі програмні продукти для ефективного проектування систем та програм з урахуванням заданих вимог до їх якісних характеристик - Здійснювати оптимізацію проектних рішень з застосуванням комп'ютерного моделювання; - Проводити аналіз поведінки систем та програм в критичних ситуаціях, оцінювати їх надійність, виконувати пошук помилок.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	к.т.н., доцент Волокита А.М., асистент Іваніщев Б.В

Освітній компонент з 10Ф-каталогу

Дисципліна	Імітаційне моделювання
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін «Програмування», «Теорія ймовірностей та математична статистика», «Емпіричні методи програмної інженерії».
Що буде вивчатися	Методи імітаційного моделювання, імітаційні моделі, програмні пакети імітаційного моделювання.
Чому це цікаво/треба вивчати	Коли явища в системі занадто складні й різноманітні, аналітична модель стає лише грубим наближенням. У таких умовах дослідник змушений

	застосовувати імітаційне моделювання. В імітаційній моделі поведінка системи описується за допомогою набору алгоритмів із застосуванням певної мови програмування, які реалізують ситуації, що виникають у реальній системі. Імітаційні моделі характеризуються точним відображенням певного процесу або явища. Тому математичні моделі, які входять до їх складу, виявляються досить складними, у них присутні нелінійні й стохастичні залежності й змінні.
Чому можна навчитися (результати навчання)	В результаті вивчення навчальної дисципліни студент повинен знати: - основні види імітаційних моделей, принципи імітаційного моделювання, визначення та характеристики базової випадкової величини, приклади базових датчиків, основні методи імітації випадкових величин з дискретними та абсолютно неперервними розподілами, принципи моделювання випадкових векторів та процесів, зокрема, систем масового обслуговування; - методи верифікації результатів моделювання. вміти: - моделювати значення базової випадкової величини, проводити верифікацію отриманих результатів моделювання, моделювати вибірки заданих об'ємів для випадкових величин з дискретними та абсолютно неперервними розподілами, моделювати значення випадкових векторів та процесів; - застосовувати імітаційні моделі для прогнозу та узгодженості з реальними статистичними даними.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Працювати з основними імітаційними моделями та застосувати отримані знання до вирішення прикладних задач.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни (з РСО), опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Протоколи і алгоритми електронного голосування
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: алгебра, дискретна математика, теорія інформації та кодування, комп'ютерні мережі, безпека інформаційних систем.
Що буде вивчатися	Протоколи і алгоритми електронного голосування для забезпечення надійного, достовірного (виключає фальшування), безпечного і транспарентного для всіх сторін і учасників (забезпечує достовірну перевірку), проведення і фіксації волевиявлення на виборах будь-якого рівня (від виборів старости студентської групи і представників у органи студентського самоврядування, до виборів народних депутатів і президента країни). Міжнародний досвід та вітчизняні розробки.
Чому це цікаво/треба	Будь-які вибори, де приймають участь три зацікавлені сторони: виборці, кандидати, третя (довірена?) сторона – організатор виборів, мають нарізно

вивчати	спрямовані вектори своїх інтересів, які перетинаються лише у одній точці – провести вибори. Один із можливих сценаріїв: виборці і кандидати мають прийняти обраним кандидата, оголошеного організатором виборів, але об'єктивно вони зацікавлені у різному: виборці воліють прийняти обраним того кандидата, за якого віддана більшість голосів, а кожен кандидат воліє, щоби оголосили обраним саме його. Організатор виборів більш за все зацікавлений оголосити вибори такими, що відбулися, щоби унеможливити претензії до себе. Досвід останніх десятиліть демонструє і більш складні сценарії і зацікавленості сторін. Як забезпечити справедливе, чесне голосування? Справедливий, неупереджений підрахунок голосів? Уникнути ефектів соціальної інженерії, корупційних чи просто агресивних впливів на виборців, кандидатів і організатора виборів? Шукаємо разом!
Чому можна навчитися (результати навчання)	Створенню, аналізу, обґрунтуванню вибору Протоколів і алгоритмів електронного голосування для забезпечення надійного, достовірного (виключає фальшування), безпечного і транспарентного для всіх сторін і учасників (забезпечує достовірну перевірку), проведення і фіксації волевиявлення на виборах будь-якого рівня (від виборів старости студентської групи і представників у органи студентського самоврядування, до виборів народних депутатів і президента країни). Шляхи створення і налаштування відповідного ПЗ.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Створювати, аналізувати, обґрунтовувати вибір та застосовувати Протоколи і алгоритми електронного голосування для забезпечення надійного, достовірного (виключає фальшування), безпечного і транспарентного для всіх сторін і учасників (забезпечує достовірну перевірку), проведення і фіксації волевиявлення на виборах будь-якого рівня (від виборів старости студентської групи і представників у органи студентського самоврядування, до виборів народних депутатів і президента країни). Створювати і налаштовувати відповідне ПЗ.
Інформаційне забезпечення	Силабус, конспект лекцій, перелік питань і завдань до семестрового контролю, навчальний посібник до виконання лабораторних робіт, завдання до виконання лабораторних робіт, завдання до самостійної роботи студентів та методичні вказівки до їх виконання.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	к.т.н., доцент Полторак В.П.

Дисципліна	Реактивне програмування
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання з об'єктно-орієнтованого програмування, паралельного програмування, багатопоточності, та функціонального програмування
Що буде вивчатися	<ul style="list-style-type: none"> • Елементи сучасних мов програмування для підтримки функціонального програмування, зокрема лямбда-виразів • Елементи сучасних мов програмування для реалізації асинхронного програмування • Визначення та базові концепції реактивного програмування • Елементи сучасних мов програмування для реалізації асинхронного програмування Використання реактивного програмування для побудови серверних систем
Чому це цікаво/треба вивчати	Реактивне програмування – одна із концепцій, що зараз активно

	розвиваються із реалізацією у всіх провідних мовах програмування
Чому можна навчитися (результати навчання)	Визначати компоненти програмних систем, де можна ефективно використовувати асинхронне та реактивне програмування. Використовувати сучасні підходи функціонального, асинхронного та реактивного програмування для розробки програмного забезпечення.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання та вміння дозволять використовувати сучасні засоби асинхронного та реактивного програмування для розробки високонавантажених ефективних серверних систем
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, контрольні завдання, презентації для лекцій
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	асист. Докука О.

Дисципліна	Мережеве управління та протоколи
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання курсів: «Основи програмування», «Теорія алгоритмів», «Основи комп'ютерних мереж», «Інтернет-технології».
Що буде вивчатися	<ul style="list-style-type: none"> • Види мереж. • Процес організації автоматизованих систем. Мережеві протоколи.
Чому це цікаво/треба вивчати	Сучасність надає можливість швидкого та якісного використання інформаційно-технологічних мереж на базі протоколів.
Чому можна навчитися (результати навчання)	Якісне використання ресурсів в комп'ютерних мережах, універсальне застосування технологій на базі мережевих протоколів, навчитися роботі в команді.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> • володіння знаннями про типологію та особливості мережі • Побудова локальних та глобальних комп'ютерних мереж; • Застосування протоколів в комп'ютерних мережах Застосування базових технологій розробки мережевих протоколів на базі стандартів еталонних моделей OSI/ISO
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, онлайн документація та альтернативні онлайн лекції
Форма проведення занять	Лекції, лабораторні роботи (комп'ютерний практикум)
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	

Дисципліна	Технології DevOps (авторський курс компанії SoftServe)
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Початкові знання з баз даних та адміністрування Linux
Що буде вивчатися	<ul style="list-style-type: none"> • Databases. overview. creation.

	<ul style="list-style-type: none"> • Simple SQL Queries. Relationships between tables. Data aggregation. Subqueries. Data Manipulation. • Data integrity. Database normalization. • Views, stored procedures and triggers. • Network. OSI model. DNS. TCP/IP Utilities. Cisco Packet Tracer. DHCP Server Configuring • Linux. Bash. • Docker. Container Automation. Docker Compose <p>Algorithms. Python usage for DevOps Tasks: work with OS, network, packages, databases, virtualization</p>
Чому це цікаво/треба вивчати	DevOps є високозатребуваним серед роботодавців напрямом. Курс надасть необхідні базові знання з тем, що будуть корисні для опанування професії ДевОпс інженера
Чому можна навчитися (результати навчання)	Контейнеризація, Розгортання інфраструктури веб-додатку
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті навички будуть корисними для здобуття професії ДевОпс інженера, а також використовуватись у роботі розробника, або тестувальника веб-додатків
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, практичні завдання
Форма проведення занять	Лекції та практичні завдання (лабораторні)
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	Представник компанії SoftServe

Дисципліна	Побудова IP-мереж на базі обладнання Juniper
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання з дисциплін: комп'ютерні мережі, інтернет-технології
Що буде вивчатися	<p>Архітектура пристроїв Juniper та операційної системи Junos. Командна стрічка операційної системи Junos. Базове конфігурування та технічне обслуговування пристроїв Juniper. Використання пристроїв Juniper для побудови маршрутизованих мереж. Статична маршрутизація. Налаштування статичної маршрутизації на маршрутизаторах Juniper. Протоколи маршрутизації OSPF та BGP. Налаштування протоколів маршрутизації OSPF та BGP на маршрутизаторах Juniper. Керування розповсюдженням маршрутної інформації. Політики маршрутизації та маршрутизація на основі фільтрів. Фільтрація трафіку з метою забезпечення безпеки мережі. Налаштування GRE та інших тунелів. Забезпечення якості обслуговування. Забезпечення надлишковості та відмовостійкості маршрутизованих мереж на базі обладнання Juniper. Використання комутаторів Juniper. Віртуальні локальні мережі (VLAN). Маршрутизація між VLAN. Забезпечення безпеки мереж Ethernet на базі комутаторів Juniper. Забезпечення надлишковості та відмовостійкості мереж Ethernet на базі комутаторів Juniper. STP, LAG, RTG.</p>
Чому це цікаво/треба вивчати	Juniper Networks – один із лідерів з виробництва мережевого обладнання. Juniper з кожним роком застосовується дедалі більше. При цьому система команд Juniper не схожа на систему команд інших виробників. Це відкриває

	нові можливості, але ускладнює вивчення обладнання. Знання Juniper – це конкурентна перевага при працевлаштуванні, в тому числі за кордоном. Крім того це цікаво. Під час навчання використовується реальна продукція Juniper Networks. Надаються навчальні матеріали. Ви зможете вивчити обладнання Juniper та підготуватися до сертифікації JNCIA-JUNOS та JNCIS-ENT.
Чому можна навчитися (результати навчання)	Проектувати комп'ютерні мережі на базі обладнання Juniper. Конфігурувати маршрутизатори та комутатори Juniper в корпоративних мережах. Підготуватися до сертифікаційних іспитів JNCIA-Junos та JNCIS-ENT
Як можна користуватися набутими знаннями і уміннями (компетентності)	Проектувати маршрутизовані та комутовані мережі на базі обладнання Juniper. Виконувати базові налаштування, моніторинг та технічне обслуговування обладнання Juniper. Налаштовувати статичну маршрутизацію на маршрутизаторах Juniper, в т.ч. агреговані та генеровані маршрути. Керувати розповсюдженням маршрутної інформації та налаштовувати балансуювання навантаження. Розуміти логіку роботи протоколів маршрутизації OSPF, BGP та налаштовувати OSPF та BGP на маршрутизаторах Juniper. Розуміти концепцію тунелювання трафіку. Налаштовувати GRE та IP-IP тунелі. Розуміти та налаштовувати різні можливості відмовостійкості (high availability) Використовувати комутатори Juniper в локальних мережах. Конфігурувати віртуальні локальні мережі (VLAN) та маршрутизацію між VLAN на базі обладнання Juniper. Налаштувати засоби безпеки в локальних мережах. Налаштувати фільтрацію трафіку на пристроях Juniper.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, електронний конспект та презентація
Форма проведення занять	Лекції, практичні роботи з обладнанням Juniper
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	к.т.н., Миронов Дмитро Вікторович JNCIA, JNCIS, CCAI, CCNA, CCNP

Дисципліна	<i>Системне програмування в середовищі Unix</i>
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Вміння програмувати на C, або C++, або Rust та вміння працювати у Unix-подібній системі на рівні користувача. Базові знання по дисциплінам Програмування, Структури даних та алгоритми.
Що буде вивчатися	Проектування та розробка системних програм для Unix-подібних систем. Програмування відбувається на рівні системних викликів, тобто на низькому рівні взаємодії з ядром. Ґрунтовна інформація по API в Unix-подібній системі (з розумінням реалізацій системних викликів в ядрі) для розробки системних програм. Дисципліна не орієнтована на системне програмування в якійсь одній конкретній реалізації Unix-подібної операційної системи, тобто буде вивчатися переносне системне програмування. Курс складається з наступних тем: середовище процесу, управління процесами, просунута

	робота з файлами, робота з сигналами, міжпроцесова взаємодія (IPC), просунуте I/O, програмування терміналів.
Чому це цікаво/треба вивчати	Цю дисципліну доцільно вивчати тим, хто буде розробляти системні програми для Unix-подібних систем. Програмування завдань в дисципліні виконується на мові C, або C++, або Rust, але набуті знання будуть корисні для розробки системних програм для Unix-подібних систем на інших мовах програмування.
Чому можна навчитися (результати навчання)	Розробляти системні програми для Unix-подібних систем на мові програмування C, або C++, або Rust для управління процесами, роботи з файлами, сигналами, використання просунутого I/O, міжпроцесової взаємодії (IPC), роботи з терміналами.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання можливо використовувати при проектуванні та розробці системних програм для Unix-подібних систем, для супроводження вихідного коду наявних системних програм для Unix-подібних систем, при розробці більш ефективних прикладних програм.
Інформаційне забезпечення	Силабус, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	ст. викладач Сімоненко А.В.

Дисципліна	Системи автоматизованого проектування комп'ютерних систем
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська, англійська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з вищої математики, програмування, дискретної математики, комп'ютерної схематехніки, архітектури комп'ютерів.
Що буде вивчатися	— основи будови систем автоматизованого проектування (САПР) комп'ютерних систем (КС); — технологія проектування спеціалізованих конвейерних КС на базі програмованих логічних інтегральних схем (ПЛІС) з використанням сучасних САПР; — проектування спеціалізованих багатопроцесорних КС та їх інтерфейсів.
Чому це цікаво/треба вивчати	— сучасні ПЛІС є чи не єдиною елементною базою, що дає змогу проектувати новітні КС, які відрізняються високими характеристиками продуктивності, надійності, енергоспоживання; — ПЛІС використовуються в медичинських приладах, телекомунікаціях, транспорті, датацентрах, інтернеті речей, військовій електроніці, аерокосмічній галузі, для вирішення задач штучного інтелекту; — є великий дефіцит професійних кадрів, як в Україні, так і в світі, які спроможні проектувати КС на ПЛІС, зарплатня яких, як правило, є вищою за зарплатню пересічного програміста.
Чому можна навчитися (результати навчання)	— складати паралельні алгоритми високопродуктивних обчислень, — програмувати алгоритми мовою VHDL для реалізації в апаратних КС, — моделювати і тестувати проекти КС, компілювати опис КС у прошивку ПЛІС.
Як можна користуватися набутими знаннями і уміннями (компетентності)	— проводити аналіз придатності ПЛІС для виконання комп'ютерних задач, які потребують високої продуктивності, — проектувати високопродуктивні КС на ПЛІС за ефективною технологією,

	<p>— виконувати модернізацію існуючих високопродуктивних КС на ПЛІС,</p> <p>— організувати виконання алгоритмів, описаних мовами С, Python, на апаратних прискорювачах на ПЛІС.</p> <p>Володіння технологією програмування ПЛІС затребуване в Україні в галузях проектування військової техніки, телекомунікаціях, захисту інформації, авіабудування, в т.ч. проектування дронів і за кордоном у цих галузях, а також в системах штучного інтелекту, IoT, ADAS, робототехніці, проектуванні замовлених мікросхем.</p>
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	д.т.н., професор Сергієнко А.М.

Освітній компонент з ІІФ-каталогу

Дисципліна	Проектування мікропроцесорних систем на базі Arduino та Raspberry PI
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: Базові знання мов програмування, базові знання електротехніки, роботи в мережі інтернет.
Що буде вивчатися	Проектування мікропроцесорних систем на базі Arduino : Цифрові порти введення-виведення, широко-імпульсна модуляція (ШІМ), робота з аналоговими датчиками, робота з серво та кроковим двигунами, робота з LCD - дисплеєм. Проектування мікропроцесорних систем на базі міні-комп'ютерів RaspberryPi .
Чому це цікаво/треба вивчати	Мікропроцесорні системи призначені для автоматизації обробки інформації та управління різними процесами. Застосування мікропроцесорних систем практично у всіх електричних пристроях - найважливіша риса технічної інфраструктури сучасного суспільства. МПС на основі МК використовуються найчастіше як вбудовані системи для вирішення завдань управління деяким об'єктом.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> ▪ Проектувати мікропроцесорні системи на базі Arduino ; ▪ Працювати з цифровими портами введення-виведення; ▪ Працювати з аналоговими датчиками та серво і кроковим двигунами; Проектувати мікропроцесорні системи на базі RaspberryPI ;
Як можна користуватися набутими знаннями і уміннями (компетентності)	При проектуванні мікропроцесорних систем на базі Arduino. При розробці систем автоматизації та робото-технічних систем на базі МПС. При проектуванні мікропроцесорних систем на базі RaspberryPI.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс, Arduino IDE, Tinkercad, OS Debian, Wiringpi
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	к.т.н., доцент Голубев Л.П.

Дисципліна	Інфраструктура програмного забезпечення WEB-застосувань
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: основ програмування, реляційні бази даних та основи SQL (можливе одночасне вивчення), основи HTML та CSS (можливе одночасне вивчення). Бажано основи JavaScript (не критично).
Що буде вивчатися	Способи аналізу функціональних вимог для сучасних програмних систем на базі мікроархітектури та їх розгортання; - створення проекту і Web-застосунків на основі типової платформи, як основу мікроархітектурного рішення; - розгортання і контейнеризація Web-застосунків на Docker. - створення мікроархітектурного рішення на основі Web-застосунків;

	- розгортання і контейнеризація програмної інфраструктури Web-застосунків на Docker.
Чому це цікаво/треба вивчати	За один семестр можна вивчити і отримати перший досвід основ Web-застосунків і розгортання програмної інфраструктури на основі Docker. Цікаво для тих, хто маючи знання основ програмування, HTML та CSS бажає отримати досвід розробки програмної інфраструктури, планує після університету працювати в цій сфері, або почати свій бізнес.
Чому можна навчитися (результати навчання)	Проектуванню і бізнес-аналітики основ Web-застосунку на основі типовий фреймворків. Повному циклу розробки Web-застосунку на основі мікроархітектурних рішень. Розгортанню програмної інфраструктури на основі контейнеризації, на прикладі Docker.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отриманні знання, вміння, досвід і навички дають можливість працевлаштування у відповідній професійній сфері. Також отриманий досвід дозволяє зробити перший крок щодо створення власного бізнесу.
Інформаційне забезпечення	Силабус, методичні рекомендації до виконання лабораторних робіт, інформаційна підтримка у Телеграмі (група відкривається на час занять), середовище розробки, довідкова система.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	к.т.н, доцент Катін П.Ю.

Дисципліна	Проектування та реалізація операційних систем
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Розуміння Асемблеру, вміння програмувати на C, або C++, або Rust. Базові знання по дисциплінам Програмування, Структури даних та алгоритми, Архітектура комп'ютерів, Системне програмування, Системне програмне забезпечення.
Що буде вивчатися	Принципи роботи операційних систем загального призначення. Проектування та реалізація (design and implementation) ядер операційних систем загального призначення. За основу беруться реалізації ядер Unix-подібних систем та реалізації інших типів ядер, тобто дисципліна не орієнтована на якусь одну конкретну операційну систему. Дається ґрунтовна інформація по задачам та методам їх вирішення в ядрі операційної системи на низькому рівні реалізації. Курс складається з чотирьох розділів: ядро та процеси, підтримка багатопотокових програм, файлові системи, управління пам'яттю.
Чому це цікаво/треба вивчати	Цю дисципліну доцільно вивчати тим, хто буде розробляти ядра операційних систем та їх частини. Ця дисципліна також корисна системним програмістам та прикладним програмістам для поглибленого розуміння роботи операційної системи, що дозволить розробляти більш ефективні програми.
Чому можна навчитися (результати навчання)	Підготуватися до розробки та розуміння вихідного коду частин ядер операційних систем загального призначення, що реалізують управління процесами та потоками, системні виклики, файлові системи, управління

	пам'яттю. Розробляти ефективні системні програми на прикладі створення алокатора пам'яті загального призначення, файлової системи та інших.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання можливо використовувати при проектуванні та розробці ядер операційних систем, для супроводження вихідного коду наявних ядер операційних систем, при розробці ефективних системних та прикладних програм.
Інформаційне забезпечення	Силабус, РСО, конспект лекцій, завдання та пояснення лабораторних робіт.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	ст. викладач Сімоненко А.В.

Дисципліна	Тестування та контроль якості (QA) вбудованих систем
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: ОС Linux, Програмування, Алгоритми і структури даних, Системне програмування, Операційні системи. Комп'ютерні мережі, Архітектура комп'ютерів, Архітектура процесорів, Мікропроцесорні системи.
Що буде вивчатися	<p>Основи тестування. Цілі тестування на різних стадіях життєвого циклу розроблення продукту. Моделі розроблення ПЗ. Основні рівні тестування та їх взаємозв'язки. Складання тестової документації.</p> <p>Робота з обладнанням на базі платформи BeagleBone Black. Самостійне створення вбудованої системи, налаштувати оточення. Досвід тестування вбудованої операційної системи з урахуванням налаштування тестового оточення, в рамках чого розглядаються питання:</p> <ul style="list-style-type: none"> - усунення несправностей у мережі (Network Troubleshooting); - налаштування операційної системи Linux та комп'ютерної мережі; - розвертання операційної системи (Linux kernel, U-Boot, BusyBox) для архітектури процесорів ARM Cortex A8 шляхом компіляції вихідного коду; різні способи прошивання мікросхеми процесора на плату; - виконувати тестування власного продукту на базі платформи BeagleBone Black; - використання Git та GitHub для контролю версій. <p>Перевірка функціональних та нефункціональних атрибутів програмного та апаратного забезпечення у самостійно створеній вбудованій системі. Створення тестів для перевірки та підтвердження вбудованого програмного і апаратного забезпечення відповідно до вимог клієнта.</p>
Чому це цікаво/треба вивчати	<p>В ІТ компаніях велика увага приділяється якості продукту, що випускається на ринок.</p> <p>Основні причини необхідності тестування, які обґрунтовують спеціальну підготовку спеціалістів-тестувальників:</p> <ul style="list-style-type: none"> - знаходження помилок в програмному та апаратному забезпеченні, - зниження ризику як для користувачів, так і для компанії, скорочення витрат на розробку та обслуговування, - підвищення продуктивності.

	<p>Тестування убудованого програмного та апаратного забезпечення - це відмінний підхід для гарантування безпеки в таких критичних додатках, як медичне обладнання, залізниця, авіація, транспортна промисловість тощо. Великі ІТ компанії, що займаються такими розробками, запрошують на роботу спеціально підготовлених фахівців-тестувальників. Для підготовки фахівців для тестування убудованих систем необхідно базові знання в області проектування та експлуатації убудованих систем, комп'ютерних мереж.</p> <p>Курс «Тестування та контроль якості (QA) вбудованих систем» може бути прослуханий студентами як окремих курс, так і у складі сертифікатної програми «Інженерія вбудованих систем та IoT», яка запроваджена на факультеті інформатики та обчислювальної техніки з 2021 – 2022 навчального року. Детально з сертифікатною програмою можна ознайомитися на сайті кафедри обчислювальної техніки. Перелік дисциплін сертифікатної програми:</p> <ul style="list-style-type: none"> • Технології програмування C/Embedded – 5 семестр; • Управління ІТ-інфраструктурними проектами – 6 семестр; • Технології програмування на ПЛІС (FPGA) – 6 семестр; • Тестування та контроль якості (QA) вбудованих систем – 7 семестр; • Інтелектуальні вбудовані Real Time системи – 8 семестр. <p>Курс викладає та є його автором сертифікований QA Engineer та фахівець компанії GlobalLogic. В цьому курсі і в сертифікатній програмі загалом розглядаються цікаві задачі розроблені у співробітництві з менторами та тренерами компанії GlobalLogic, які також приймають участь в лекційних та практичних заняттях. Ми намагаємося допомогти студентам краще зрозуміти реалії сучасного ринку Embedded Systems та потреби роботодавців, подолати важкий поріг входження в Embedded та світ HW.</p> <p>Дефіцит фахівців о області Embedded Systems та IoT на сьогодні доволі значний, тому опанування дисциплін сертифікатної програми дозволить розширити можливості студентів в пошуку майбутньої роботи та підвищить його конкурентоспроможність. З точки зору компанії GlobalLogic участь в сертифікатній програмі дозволить сформувати необхідні знання та вміння, що задовольнять її, як роботодавця.</p>
<p>Чому можна навчитися (результати навчання)</p>	<ul style="list-style-type: none"> - Налаштовувати операційну систему Linux та комп'ютерну мережу; - Розвертати операційну систему (Linux kernel, U-Boot, BusyBox) для архітектури процесорів ARM Cortex A8 шляхом компіляції вихідного коду. - Використовувати різні способи прошивання мікросхеми процесора на платі. Користуватися різними інтерфейсами, а також програми емуляції терміналів для комунікації і налагодження убудованих пристроїв. - Виконувати тестування власного продукту на базі платформи BeagleBone Black. - Знаходити помилки в програмному та апаратному забезпеченні убудованих систем. - Усувати несправності в роботі комп'ютерної мережі. - Складати тестову документацію. - Працювати в команді. <p>Отримати базову підготовку до сертифікації QA спеціаліста по ISTQB</p>
<p>Як можна користуватися набутими знаннями і уміннями (компетентності)</p>	<p>Працювати в ІТ тестувальником програмного та апаратного забезпечення.</p> <p>Продовжувати підготовку в області розробника програмного та апаратного забезпечення убудованих систем.</p> <p>Продовжити підготовку в області фахівця по автоматизації тестування</p>

	програмного і апаратного забезпечення. Отримані базові знання в області основ QA тестування можуть бути використані в області тестування програмного забезпечення, а також бути основою для самостійної підготовки до сертифікації QA спеціаліста по ISTQB
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, практичні та семінарські заняття, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	д.т.н., професор Клименко І.А., асистент Таранюк В.А.

Дисципліна	Методика та організація наукових досліджень
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Теорія ймовірності, Алгоритми та методи обчислень, Системне програмування, Системне програмне забезпечення, Комп'ютерні системи, Структури даних і алгоритми, Організація баз даних, Програмування, Архітектура комп'ютерів, Комп'ютерні мережі, Дискретна математика
Що буде вивчатися	- Методи проведення наукових досліджень - Методи математичного моделювання - Методи відображення та перетворювання інформації
Чому це цікаво/треба вивчати	Дисципліна закладає основи самостійної наукової роботи починаючи від лабораторних робіт, курсового та дипломного проектування
Чому можна навчитися (результати навчання)	Створювати різноманітні моделі, методик виконання наукових досліджень
Як можна користуватися набутими знаннями і вміннями (компетентності)	Планувати наукові дослідження, обробляти результати наукових досліджень, обробляти та аналізувати отримані результати
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	к.т.н., доцент Селіванов В.Л.

4 курс 8 семестр

Освітній компонент з І2Ф-каталогу

Дисципліна	Алгоритми та математичні методи захисту інформації
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська

Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін «Програмування», «Вища математика», «Захист інформації».
Що буде вивчатися	Алгоритми та математичні методи захисту інформації.
Чому це цікаво/треба вивчати	Криптографія - це наука, яка займається будівництвом безпечних шифрів, тобто алгоритмів, що забезпечують перетворення електронних документів в зовні випадковий набір символів, з яких можна відновити вихідний документ, тільки знаючи якийсь пароль (секретне слово). Криптографія розвивалася з самого початку історії людства. Відомі приклади шифрування текстів з глибокої давнини - у римлян, греків та інших народів. Проте ера сучасної криптографії почалася порівняно недавно, в 70-і роки ХХ століття. В межах даної навчальної дисципліни розглядається учбовий матеріал, що має відношення до основ побудови сучасних алгоритмів криптографії та пов'язаний з ними.
Чому можна навчитися (результати навчання)	Студент повинен після засвоєння матеріалу навчальної дисципліни: знати: - концепції інформаційної безпеки, принципів безпечного проектування ІС а ІТ, методології безпечного програмування, погроз і атак, безпеки комп'ютерних мереж, методи криптографії. вміти: - зберігати конфіденційність, цілісність та доступність інформації, забезпечувати автентичність, відстежуваність та надійність інформації в умовах неповноти та невизначеності вихідних даних, багатокритеріальності професійних задач.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти та експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни (з РСО), опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Кібербезпека комп'ютерних мереж
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання з дисциплін: комп'ютерні мережі, інтернет-технології
Що буде вивчатися	Базові поняття кібербезпеки. Різновиди шкідливого ПЗ. Інструменти вивчення обчислювальних систем та мереж. Різновиди атак. Управління ризиками. Ланцюг кібератаки. Елементи ланцюга кібератаки та засоби протидії. Використання стандартних мережевих утиліт для вивчення мережі. Сканування портів. nmap, zenmap. Розширені можливості nmap для вивчення мережі. Сканування вразливостей. Протидія скануванню вразливостей. Використання сканерів вразливостей: XSpider, Nessus, OpenVAS.

	<p>Вразливості мережевих протоколів та їх використання. Засоби моделювання мережі. Віртуальні мережеві пристрої. Використання систем віртуалізації VMware Player/Workstation, ESXi, VirtualBox. Використання системи емуляції EVE-NG. Моніторинг та фільтрація трафіка. Stateless и Stateful файєрволи. Системи виявлення та запобігання вторгненням. Системи Advanced Malware Protection и Advanced Threat Protection Концепція системи безпеки Junos. Сучасні завдання та виклики для системи безпеки. Дизайн системи безпеки. Мережеві екрани Juniper SRX. Початкова конфігурація. Об'єкти системи безпеки. Security Zone, Security Screen, Security Address, Security Service. Створення об'єктів системи безпеки. Політики безпеки. Міжзонні та глобальні безпекові політики. Шлюз рівня додатків із уніфікованими політиками безпеки. Створення політик безпеки на Juniper SRX. Сервіси безпеки. Додавання сервісів IDP та служби ідентифікації користувачів до політик безпеки UTM. Контентна фільтрація. Web-фільтрація. Робота з антивірусом. Модуль Antispam. Хмарний сервіс SkyATP. Огляд SkyATP. Блокування загроз. Трансляція адрес (NAT). IPsec VPNs. Побудова тунелю IPsec. Обробка трафіку в тунелі IPsec. IPsec тунель "точка-точка" Підвищення безпеки мережевих пристроїв Juniper. Вразливості локальних мереж. Підвищення безпеки локальних мереж за допомогою комутаторів Juniper.</p>
Чому це цікаво/треба вивчати	<p>Знання кібербезпеки та сучасного мережевого обладнання є суттєвою конкурентною перевагою на ринку праці. В курсі розглядаються основи кібербезпеки та використання обладнання Juniper SRX для забезпечення безпеки комп'ютерних мереж. Juniper – один з основних виробників мережевого обладнання. Обладнання Juniper широко використовується в Україні та за кордоном. Курс дозволяє засвоїти основи кібербезпеки, надбати знання та навички з конфігурування обладнання Juniper для забезпечення мережевої безпеки. Курс дозволяє підготуватись до сертифікації JNCIA-SEC.</p>
Чому можна навчитися (результати навчання)	<p>Розумінню основних понять кібербезпеки. Використанню стандартних мережевих утиліт та спеціалізованого ПЗ для вивчення мережі. Використанню мережевих екранів Juniper SRX для запобігання мережевих атак. Підготуватися до сертифікаційного іспиту JNCIA-SEC.</p>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Розуміти основні принципи кібербезпеки. Використовувати стандартне та спеціалізоване ПЗ для вивчення мережі, пошуку та аналізу вразливостей. Виконувати конфігурування мережевих екранів Juniper SRX. Налаштувати фільтрацію трафіку та запобігання вторгнень на базі Juniper SRX. Використовувати систему Sky ATP. Налаштувати NAT. Налаштувати IPsec VPN. Використовувати комутатори Juniper для підвищення безпеки локальної мережі.</p>
Інформаційне забезпечення	<p>Навчальна і робоча програми дисципліни, електронний конспект та презентація</p>
Форма проведення занять	<p>Лекції, практичні роботи</p>

Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	к.т.н., Миронов Дмитро Вікторович JNCIA, JNCIS, CCAI, CCNA, CCNP

Дисципліна	Управління виробництвом на базі застосування систем класу ERP
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Відомості про застосування інформаційних систем в бізнесі Відомості про виробничі бізнес системи
Що буде вивчатися	<ul style="list-style-type: none"> * Теоретичні та практичні аспекти побудови виробничих бізнес-систем * Стандарти управління виробництвом: MRP-II, LeanProduction, TOC Практика застосування інформаційних систем для автоматизації процесів у виробництві.
Чому це цікаво/треба вивчати	Виробничі системи як напрямок автоматизації завжди будуть знаходитись на передньому краї застосування ІТ-рішень в бізнесі. Сучасні стандарти управління виробництвом є основою побудови ІС у вигляді ERP-додатків
Чому можна навчитися (результати навчання)	Матеріали курсу поєднують бізнес дослідження в сфері управління виробничими системами з аналітичними ІТ-дослідженнями побудови ERP-систем. Зрозуміння цих моментів створює фундаменальний базис для використання ІТ-технологій програмування та інформаційного проєктування на практиці.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробники програмного та інформаційного забезпечень можуть виконувати функції та роботи аналітичного характеру, приєднуватись до практичної роботи на більш ранніх стадіях виконання ІТ проєктів.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, онлайн документація
Форма проведення занять	Лекції та лабораторні
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	

Дисципліна	Комп'ютерна лінгвістика
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	<ul style="list-style-type: none"> - Базові знання програмування - Базові знання теорії ймовірності та мат. статистики
Що буде вивчатися	<ul style="list-style-type: none"> - Види аналізу тексту - Математичне та інформаційне забезпечення лінгвістичного аналізу - Методи визначення параметрів тексту та особливих рис авторів
Чому це цікаво/треба вивчати	На теперешній час та надалі комп'ютерна лінгвістика буде відігравати все більш значну роль при аналізі інформаційних потоків.

	<p>Використання на базі методів комп'ютерної лінгвістики методів машинного навчання, забезпечення вхідної інформації для роботи нейронних мереж, різноманітність задач, що потребують вирішення - все це робить дану дисципліну однією з найзатребуваних для сучасного ІТ-спеціаліста, що спрямовує свою діяльність на роботі з інформацією.</p> <p>Коло задач комп'ютерної лінгвістики коливається від пошуку подібних документів (задачі пошукових систем, плагіату тощо), визначення тональності текстів для аналізу вподобань (цільова реклама, системи відгуків, соціальні науки) до виявлення прихованих рис, що дозволяють визначити параметри автору тексту, вилучення фактажу з тексту для аналізу, побудови психологічного портрету автора, аналізу соціальних мереж та медіа.</p>
Чому можна навчитися (результати навчання)	<p>1) виділення рівнів аналізу тексту, типовим методам вирішення практичних задач, особливостям використання різних методів аналізу;</p> <p>2) методам вирішення типових задач, розбиття (декомпозиції) складних задач аналізу тексту.</p>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>- використовувати методи та підходи комп'ютерної лінгвістики для вирішення практичних задач;</p> <p>- використовувати та досліджувати методи збору інформації, фільтрації інформації.</p> <p>- аналізувати інформаційні потоки на предмет емоційного забарвлення</p>
Інформаційне забезпечення	навчальна та робоча програма дисципліни, РСО, навчальний посібники та підручники (електронні видання)
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	к.т.н, доцент Фіногенов О.Д.

Дисципліна	Основи науки про дані
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з вищої математики (математичний аналіз, лінійна алгебра та аналітична геометрія), теорія ймовірностей та математична статистика, програмування
Що буде вивчатися	Загальні поняття про сутність аналізу даних, застосування принципу Бонферроні, методу пошуку найближчих сусідів, шинглінгу документів. Принципи збереження схожості скорочених наборів, локально-чутливе хешування та його застосування, міри відстані. Поняття про аналіз потоків даних різної природи та складності та аналіз посилань. Модель ринкової корзини та алгоритм A-Priori
Чому це цікаво/треба вивчати	Наука про дані має вагомим практичне значення для вирішення актуальних повсякденних проблем. Застосування науки про дані змінює спосіб життя, праці й мислення, оскільки однією з умов успішного розвитку світової економіки на сучасному етапі є можливість фіксувати й аналізувати накопичені набори даних. Сучасні ефективні методи роботи з даними сприяють як

	підвищенню конкурентоспроможності різних сфер економічної діяльності, так і фахівців, які володіють цими методами.
Чому можна навчитися (результати навчання)	Створювати моделі для даних з застосуванням узагальнення та виділення ознак. Використовувати методи обчислення жакартової подібності множин. Застосовувати міри відстані для визначення подібності, зокрема, косинусу відстань, відстань редагування, відстань Хеммінга, локально-чутливі функції. Аналізувати потоки даних шляхом побудови моделі поточкових даних та фільтрації потоків, здійснювати підрахунок різних елементів у потоці, зокрема, підрахунок у вікні та згасаючих вікнах, використовувати алгоритми DGIM. Здійснювати аналіз посилань з застосуванням PageRank, його ефективне обчислення, обчислювати тематично чутливий PageRank, визначати спам посилань.
Як можна користуватися набутими знаннями і уміннями (компетентності)	- для формування початкових наборів даних з застосуванням принципу Бонферроні; - для пошуку текстуально подібних документів (плагіату, дзеркальних сторінок, статей з одного джерела); - для визначення подібності он-лайнних покупок та рейтингу фільмів; - для розпізнавання об'єктів, співставлення відбитків пальців та газетних статей; - для створення програмних систем ефективною фільтрації небажаних елементів потоку (вхідні потоки електронної пошти); - для створення телепорт-наборів.
Інформаційне забезпечення	Навчальна і робоча програма дисципліни, РСО, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	д.т.н., професор Новотарський М.А.

Освітній компонент з ІЗФ-каталогу

Дисципліна	Технології штучного інтелекту
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: математичний аналіз, дискретна математика, об'єктно – орієнтоване моделювання, теорія алгоритмів, теорія автоматичного управління, моделювання систем, веб-технології та веб дизайн, програмування.
Що буде вивчатися	Поняття штучного інтелекту. Поняття інтелектуальної системи (ІС) та інтелектуальної задачі (ІЗ). Способи подання інтелектуальної задачі та методи пошуку рішень. Пошук рішень ІЗ у просторі станів. Знання та моделі представлення знань, продукційні моделі представлення знань у СШ. Управління пошуком рішень у продукційних системах. Класичні технологій ШІ (машинне навчання, глибоке навчання, компютерний зір, обробка природньої мови). Експертні системи (ЕС): призначення та принципи побудови; узагальнена архітектура; класи задач, які вирішуються за

	допомогою ЕС. Розробка ЕС: етапи розробки; придбання знань; пошук та пояснення рішень. Інженерія знань. Онтологічний підхід до представлення та інтеграції знань у розподілених інформаційних середовищах типу Інтернет.
Чому це цікаво/треба вивчати	Технології штучного інтелекту використовуються практично у всіх сферах, включаючи ІТ. Фахівці здатні вирішувати складні задачі, пов'язані зі створенням та навчанням штучних нейронних мереж, розпізнанням мови та зображень, аналізу великого обсягу даних, управляти складними системами і системами різного призначення.
Чому можна навчитися (результати навчання)	Результатами вивчення дисципліни є знання: основних підходів до вирішення інтелектуальних задач, основних понять та означень штучного інтелекту, технологій, підходів і методів штучного інтелекту, способів подання інтелектуальної задачі та методів пошуку рішень, моделей представлення знань у системах із ШІ, сучасних тенденції та підходів до створення систем із ШІ.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Студенти можуть досліджувати й оцінювати програмні продукти з алгоритмами штучного інтелекту, розробляти нейромережі різної архітектури, зокрема: перцептрон, неокогнітрон, когнітрон, мережі Хопфілда та Хемінга, розробляти системи розпізнавання образів, застосовувати класичні технології штучного інтелекту для вирішення інтелектуальних задач, модулювати структуру та розробляти експертні системи.
Інформаційне забезпечення	Конспект лекцій - https://campus.kpi.ua/student/index.php?mode=mob&show&irid=202438 Презентації - https://campus.kpi.ua/student/index.php?mode=mob&show&irid=202439 силабус курсу, методичні вказівки по виконанню лабораторного практикуму, контрольні завдання.
Форма проведення занять	Лекції та комп'ютерні практикуми
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	ст. викладач Польшакова О.М.

Дисципліна	Програмування вбудованих систем на основі архітектури ARM
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання: основ програмування (C, C++ або C#) і алгоритмізації. Бажані знання основ мікроелектроніки, дискретної математики.
Що буде вивчатися	Способи аналізу функціональних вимог, структури і технології вбудованих систем, приклади використання у автомобільній індустрії; - структурна, функціональна схема вбудованої системи на базі мікроконтролера (у першому наближенні ARM, Cortex-M (STM32F10x)); - види сучасних програмних технологій програмування C (C++); - досконале вивчення обраної технології програмування, для студентів з досвідом дозволено самостійно обирати технологію; - операційні системи реального часу для вбудованих систем; - вбудована система як елемент автомобільного автопілоту; - основи бізнесу у індустрії вбудованих систем.
Чому це цікаво/треба вивчати	За один семестр можна вивчити базові основи програмування і розробки вбудованих систем. Отримати перший досвід, визначитись з потенційним

	<p>напрямок роботи у цій сфері:</p> <ul style="list-style-type: none"> - бізнес-аналітик у сфері розробки вбудованих систем, у тому числі у автомобільній сфері; - інженер-розробник програмної і апаратної складових вбудованих систем, у тому числі у автомобільній сфері; - тестер (проектувальник вбудованих систем), та інші споріднені професійні напрямки. Визначитись з питаннями початку власного бізнесу.
Чому можна навчитися (результати навчання)	<p>Основний напрям - програмування вбудованої системи, розробка програмної частини</p> <p>При цьому надаються вміння, досвід і навички проектування і бізнес-аналітики вбудованих систем (на прикладі автомобільної сфери).</p> <p>Повному циклу розробки вбудованих систем.</p> <p>Самостійному і набуттю знань з питань проектування і програмування вбудованої системи.</p>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Отриманні знання, вміння, досвід і навички дають можливість працевлаштування у відповідній професійній сфері.</p> <p>Також отриманий досвід дозволяє зробити підґрунтя щодо створення власного бізнесу у галузі вбудованих систем.</p>
Інформаційне забезпечення	<p>Силабус, методичні рекомендації, підтримка у Телеграмі (група відкривається на час занять). Лабораторія ауд. 517-18, що має все для розробки (у першому наближенні ARM, Cortex-M (приклад STM32F10x)), середовище розробки https://www.keil.com.</p>
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	к.т.н., доцент Катін П.Ю.

Дисципліна	Інженерія знань
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Дисципліна “Інженерія знань” базується на наступних дисциплінах - “Дискретна математика”, “Теорія ймовірностей”, “Функціональне програмування”, “Мультипарадигмене програмування”. З практичного боку бажані знання мов штучного інтелекту LISP, PROLOG, системи CLIPS.
Що буде вивчатися	<ul style="list-style-type: none"> – загальні методи представлення знань у системах штучного інтелекту, починаючи з попереднього досвіду класичної логіки та доведення теорем, потім звертається до нових підходів, які поширюють класичну логіку - наприклад, для обробки якісної або невизначеної інформації - та для покращення її обчислювальної придатності; – розглядаються особливі проблеми репрезентації та аргументування деяких основних сфер знань, включаючи час, простір, причинно-наслідкові зв'язки та дії; <p>досліджуються важливі програми подання знань та міркувань, області застосування яких охоплюють широкий спект штучного інтелекту, включаючи відповіді на запитання, семантичні мережі, планування, робототехніку та мультиагентні системи.</p>
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> • представлення знань та міркування лежить в основі великого виклику штучного інтелекту; <p>зрозуміти природу інтелекту та пізнання настільки добре, що комп'ютери можуть бути створені для виявлення людських здібностей.</p>

Чому можна навчитися (результати навчання)	– застосовувати існуючі методи та алгоритми представлення знань для розв’язання практичних задач штучного інтелекту застосовувати методи розробки алгоритмів, конструювання експертних систем та структур знань
Як можна користуватися набутими знаннями і уміннями (компетентності)	– застосовувати моделі інженерії знань для розробки програмного забезпечення систем, що базуються на знаннях (knowledge base systems) аналізувати, цілеспрямовано шукати і вибирати необхідні засоби інженерії знань для опису, конструювання, впровадження баз знань предметних областей
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, лабораторні завдання
Форма проведення занять	Лекції та лабораторні (комп’ютерні практикуми, розрахунково-графічна робота)
Семестровий контроль	Залік, 4 курс 8 семестр
Додаткові дані	25 тем дисципліни, організовані у три розділи «Загальні методи», «Спеціалізовані представництва» та «Застосування», забезпечують унікальний огляд найкращих результатів, котрі досягла інженерія знань. Ми сподіваємось, що студенти, дослідники та практики у всіх сферах штучного інтелекту та когнітивної науки знайдуть цей курс корисним ресурсом. Частина тем дисципліни може бути винесена на самостійне вивчення.

Дисципліна	Теорія формальних граматики
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Дисципліна “Теорія формальних граматики” базується на наступних дисциплінах - “Дискретна математика”, “Теорія ймовірностей”.
Що буде вивчатися	– основи формальних граматики як формальних систем ; – класифікація і ієрархія формальних граматики; – ймовірнісні граматики, нечіткі граматики, автоматні граматики і скінченні автомати; абстрактні машини Тьюринга, Поста; нечіткі алгоритми, нечіткі мови, нечітка машина Тьюринга
Чому це цікаво/треба вивчати	• теорія формальних граматики — це шлях до аналізу існуючих мов програмування та створення нових; використання класичних методів формальних граматики разом з сучасними підходами для реалізації нових мов програмування
Чому можна навчитися (результати навчання)	– застосовувати методи формальних граматики різних класів для опису мов, процесів, різних практичних застосувань; застосовувати методи для аналізу алгоритмів, конструювання формальних систем і моделей
Як можна користуватися набутими знаннями і уміннями (компетентності)	– застосовувати формальні граматики для аналізу і розробки програмного забезпечення конструювання програмних систем аналізувати, цілеспрямовано шукати і вибирати необхідні засоби для опису, конструювання формальних мов, формальних алгоритмів, впровадження цих підходів для формалізації процесів розв’язання складних задач
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, лабораторні завдання
Форма проведення занять	Лекції та лабораторні (комп’ютерні практикуми)
Семестровий контроль	Залік, 4 курс 8 семестр

Викладач	
Дисципліна	Розподілені системи обробки інформації
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформатики та програмної інженерії
Вимоги до початку вивчення	Базові знання курсів: «Основи програмування», «Теорія алгоритмів», «Проектування інформаційних систем», «Комп'ютерні мережі».
Що буде вивчатися	<ul style="list-style-type: none"> – Стратегії розвитку інформаційних систем та технологій – Процес організації автоматизованих систем. Базові платформи розподілених систем.
Чому це цікаво/треба вивчати	Сучасність надає можливість швидкого та якісного впровадження інформаційного-технологічного ресурсу в ІТ індустрії та у всі сфери суспільства.
Чому можна навчитися (результати навчання)	Якісне використання ресурсів в глобальних мережах, універсальне застосування технологій та баз даних, застосовувати інтерактивні веб-сервіси, навчитися роботі в команді.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> – володіння знаннями про типологію та особливості – комп'ютерних мереж; вміння працювати з базами даних, проводити розподілені обчислення
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, онлайн документація та альтернативні онлайн лекції
Форма проведення занять	Лекції, лабораторні роботи (комп'ютерний практикум)
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	

Дисципліна	Статистичні методи машинного навчання
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з вищої математики (математичний аналіз, лінійна алгебра та аналітична геометрія), теорія ймовірностей та математична статистика, програмування, навички розв'язування типових задач цих дисциплін
Що буде вивчатися	Основні методи машинного навчання для задач класифікації, кластеризації і регресії (прогнозування), зокрема, принципи побудови деяких основних класифікаторів, задачі, які потребують застосування статистичних методів машинного навчання, алгоритмічні та обчислювальні підходи до їх реалізації з поглибленим вивченням мови програмування Python та застосуванням бібліотек TensorFlow і THEANO.
Чому це цікаво/треба вивчати	Для отримання істотних конкурентних переваг у сфері ІТ-технологій, оскільки машинне навчання широко використовується при побудові пошукових машин, які навчаються надавати релевантні результати; моделей для генерування таргетованої реклами в рамках масових рекламних кампаній; анти-спамного програмного забезпечення з фільтрування

	повідомлень електронної пошти; програмного забезпеченням для виявлення шахрайства в мережі Інтернет, зокрема при транзакціях з кредитними картками; розпізнавання образів та голосових команд. Машинне навчання також широко використовується в наукових програмах, таких як біоінформатика, медицина та астрономія.
Чому можна навчитися (результати навчання)	Знання принципів побудови векторів ознак, правил рішень і класифікації; основних видів класифікаторів; принципів побудови лінійних класифікаторів; принципів побудови нелінійних класифікаторів; особливостей вибору ознак класифікації і попередньої обробки даних. Вміння вибирати відповідний вид класифікатора залежно від розв'язуваної задачі; вибирати набір ознак для класифікації і проводити попередню обробку даних; застосовувати алгоритми побудови і навчання класифікатора за вибіркою; виконувати обчислення, пов'язані з навчанням і роботою класифікатора, в середовищі Python з застосуванням бібліотек TensorFlow і THEANO.
Як можна користуватися набутими знаннями і уміннями (компетентності)	- для вибору, побудови, навчання і використання основних класифікаторів при вирішенні завдань; - для самостійної роботи в сучасних програмних комплексах; освоєння великого обсягу інформації; - для постановки задач та проведення експерименту, програмування задач аналізу даних; для візуалізації одержаних результатів.
Інформаційне забезпечення	Навчальна і робоча програма дисципліни, РСО, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	д.т.н., професор Новотарський М.А.

Дисципліна	Мережне програмування в середовищі Unix
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Вміння програмувати на C, або C++, або Rust та вміння працювати у Unix-подібній системі на рівні користувача. Базові знання по дисциплінам Програмування, Структури даних та алгоритми, Комп'ютерні мережі.
Що буде вивчатися	Розглядається проектування та розробка системних та серверних програм, що працюють з мережею для Unix-подібних систем. Програмування відбувається на рівні системних викликів, тобто на низькому рівні взаємодії з ядром. Ґрунтовна інформація по API в Unix-подібній системі (з розумінням реалізацій системних викликів в ядрі) для розробки системних та серверних програм, що працюють з мережею. Дисципліна не орієнтована на мережеве програмування в якійсь одній конкретній реалізації Unix-подібної операційної системи, тобто буде вивчатися переносне мережеве програмування. Курс складається з наступних тем: початкова робота з сокетми TCP, UDP, SCTP, робота з іменами та адресами, просунута робота з сокетми TCP, UDP, SCTP, робота з сокетми локального домену, просунутий мережевий I/O, реалізації клієнт/сервер різними способами (тема присутня в кожній іншій темі, тому не є окремою темою).
Чому це цікаво/треба вивчати	Цю дисципліну доцільно вивчати тим, хто буде розробляти системні та серверні програми, що працюють з мережею для Unix-подібних систем. Програмування завдань в дисципліні виконується на мові C, або C++, або

	Rust, але набуті знання будуть корисні для розробки мережевих програм та серверів для Unix-подібних систем на інших мовах програмування.
Чому можна навчитися (результати навчання)	Розробляти системні та серверні програм, що працюють з мережею для Unix-подібних систем на мові програмування C, або C++, або Rust, які працюють з сокетами TCP, UDP, SCTP, створювати клієнт/сервер програми різними способами.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання можливо використовувати при проектуванні та розробці системних та серверних програм, що працюють з мережею для Unix-подібних систем, для супроводження вихідного коду наявних програм, що працюють з мережею, для Unix-подібних систем.
Інформаційне забезпечення	Силабус, РСО, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	ст. викладач Сімоненко А.В.

Освітній компонент з ІФ-каталогу

Дисципліна	Інтелектуальні транспортні системи
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін «Комп'ютерні мережі», «Інтернет речей», «Дискретна математика», «Системна інженерія».
Що буде вивчатися	Теоретичні та практичні аспекти проектування, розробки, експлуатації сучасних інтелектуальних транспортних систем, їх основні задачі та функції.
Чому це цікаво/треба вивчати	Уже понад 70 років для регулювання транспортних потоків використовуються передові технології. Виробники транспортних засобів розробляють передові технології для того, щоб транспортні засоби стали безпечнішими, пересування завдавало менше стресу і було зручнішим. Передові технології все більше і більше застосовуються до великих систем громадського транспорту, а також для поширення інформації про прибуття поїздів та автобусів для пасажирів. В секторі вантажного транспорту ціла низка технологій застосовується для покращення ефективності руху транспортних засобів та відповідних комерційних операцій як ланки ланцюга постачання. Ці різні технології тепер відомі під збірною назвою інтелектуальні транспортні системи (ІТС). Відтак, ІТС можна визначити як застосування комп'ютерних, інформаційних та комунікаційних технологій для управління транспортними засобами та мережами у реальному часі, включаючи переміщення людей і товарів.
Чому можна навчитися (результати навчання)	У результаті вивчення дисципліни студенти повинні знати : - характеристики інтелектуальних транспортних систем; - структуру інтелектуальних систем; - характеристики та види інтелектуальних систем для інфраструктури транспорту; - характеристики та види інтелектуальних систем для транспортних засобів. вміти : - ідентифікувати параметри інтелектуальних транспортних систем; - описувати структуру інтелектуальних систем; - вибирати характеристики для інтелектуальних систем у сфері

	інфраструктури транспорту; - вибрати характеристики для інтелектуальних систем управління транспортними засобами.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробка та застосування елементів інтелектуальних систем на транспорті.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни (з РСО), опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Оптимізація та балансування навантажень у базах даних
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Інформаційних систем та технологій
Вимоги до початку вивчення	Базові знання з дисциплін: <ul style="list-style-type: none"> • досвід у використанні БД: вміння встановлювати, підключатися до БД, розуміння синтаксису SQL та бажано досвід у використанні PostgreSQL БД • написання запитів до БД з використанням декількох таблиць • мінімальні знання по архітектурі ПК та комп'ютерним мережам • базові знання по віртуалізації та, зокрема, робота з Docker знання алгоритмів пошуку та роботи з деревами
Що буде вивчатися	<ul style="list-style-type: none"> • вивчення різних типів БД, включно з аналізом реляційних БД та PostgreSQL БД • основні аспекти, використання БД, розбір проблеми нецільового використання окремих типів БД у IT-інфраструктурі • роль індексації у БД, створення власних індексів • CAP-теорема та її роль при проектуванні власних продуктів з урахуванням використання БД • оптимізація запитів до БД – використання процесорного часу, пам'яті та кешу • бекапи та снапшоти – задачі, що вони вирішують та аспекти їх використання • міграція БД
Чому це цікаво/треба вивчати	<p>На сьогодні більшість програмних застосунків використовують бази даних. Здебільшого розробники використовують ту БД, до якої звикли. За такого підходу не враховуються специфіка задач, що вирішують обрані БД у програмних продуктах. Мета курсу:</p> <ul style="list-style-type: none"> • продемонструвати особливості кожного типу БД (їх переваги та недоліки) • продемонструвати доречність використання окремих типів БД • навчити оптимізації запитів у реляційні БД для зменшення навантаження на сервери • навчити раціонально використовувати кеш • підвищення надійності систем, до використовують БД шляхом бекапу даних та горизонтального розширення сервісів

Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> • оптимізація запитів до БД • проектування власної ІТ-інфраструктури для вирішення задач при високому навантаженні • підвищення надійності ІТ-інфраструктур при використанні БД • балансування навантаження у ІТ-інфраструктурах на прикладі сервісу БД <p>основи міграції БД</p>
Як можна користуватися набутими знаннями і вміннями (компетентності)	<p>Після проходження курсу, студент буде:</p> <ul style="list-style-type: none"> • розуміти різницю між реляційними та нереляційними БД • вміти проектувати архітектуру ПП (програмний продукт) з урахуванням задач, що цей ПП виконує • доцільно використовувати процесорний час, кеш декількох рівнів • розуміти переваги HDD над що SSD
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	к.т.н., доцент Галушко Д.О.

Дисципліна	Інтелектуальні вбудовані (Real Time) системи
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Дискретна математика, Комп'ютерна логіка, Комп'ютерна арифметика, Програмування С, С++, Python, Комп'ютерні мережі, Системне програмування, Архітектура комп'ютерів, Архітектура процесорів, Мікропроцесорні системи. ОС Linux.
Що буде вивчатися	<p>Теоретичні та практичні основи організації, побудови та структури систем реального часу. Напрацювання практичних навичок розробки вбудованих систем та пристроїв IoT для вирішення завдань обробки даних в реальному часі на актуальній елементній базі.</p> <p>Тематика лабораторних робіт:</p> <ul style="list-style-type: none"> – прикладні задачі пересилки та обробки даних у системах реального часу; – розробка вбудованих систем та пристроїв IoT для рішення задач в реальному часі; – конфігурація та використання ОС реального часу RTOS, розробка та конфігурація програмного забезпечення для мікроконтролерів на базі процесорів ARM; – прошивка плат, розвертання ядра операційної системи Linux на процесорних ядрах ARM; розробка модулів ядра для вбудованих систем; – написання автотестів. <p>Бакалаврські проекти: Розроблення проекту з власною прошивкою на базі апаратних засобів для програмування та налагодження мікроконтролерів. Використання платформ BeagleBone Black, Raspberry Pi, STM StarterKit GlobalLogic, ESM32.</p>
Чому це цікаво/треба вивчати	Системи реального часу – це системи, які відстежують зовнішнє середовище або керують ним. Зовнішнє середовище пов'язане з комп'ютерною системою за допомогою системи датчиків та складається з фізичних або біологічних, природних або штучних об'єктів, часто навіть людей і тварин.

	<p>Система реального часу відповідає часовим та іншим обмеженням, які накладаються на неї поведінкою зовнішнього світу в реальному часі.</p> <p>На сьогодні застосування систем реального часу є повсюдним та все більше поширюються. Системи реального часу це невід'ємна частина Інтернету речей (IoT) та Інтернету всього (IoE). Комерційна, урядова, військова, медична, освітня, культурна та інші інфраструктури сьогодні пов'язані в єдину мережу IoT і є конгломератом, як автономних та і взаємозв'язаних та систем реального часу.</p> <p>Стрімкий розвиток сучасних технологій IoT потребує відповідної специфіки знань та вмінь від сучасних фахівців в області комп'ютерної інженерії, програмування та загалом інформаційних систем.</p> <p>Курс «Інтелектуальні вбудовані Real Time системи» може бути прослуханий студентами як окремий курс, так і у складі сертифікатної програми «Інженерія вбудованих систем та IoT», яка запроваджена на факультеті інформатики та обчислювальної техніки з 2021 – 2022 навчального року. Детально з сертифікатною програмою можна ознайомитися на сайті кафедри обчислювальної техніки. Перелік дисциплін сертифікатної програми:</p> <ul style="list-style-type: none"> • Технології програмування C/Embedded – 5 семестр; • Управління IT-інфраструктурними проектами – 6 семестр; • Технології програмування на ПЛІС (FPGA) – 6 семестр; • Тестування та контроль якості (QA) вбудованих систем – 7 семестр; • Інтелектуальні вбудовані Real Time системи – 8 семестр. <p>В цьому курсі і в сертифікатній програмі загалом розглядаються цікаві задачі розроблені у співробітництві з менторами та тренерами компанії GlobalLogic, які також часто приймають участь в лекційних та практичних заняттях. Ми намагаємося допомогти студентам краще зрозуміти реалії сучасного ринку Embedded Systems та потреби роботодавців, подолати важкий поріг входження в Embedded та світ HW.</p> <p>Дефіцит фахівців о області Embedded Systems та IoT на сьогодні доволі значний, тому опанування дисциплін сертифікатної програми дозволить розширити можливості студентів в пошуку майбутньої роботи та підвищить його конкурентоспроможність. З точки зору компанії GlobalLogic участь в сертифікатній програмі дозволить сформувані необхідні знання та вміння, що задовольнять її, як роботодавця.</p>
<p>Чому можна навчитися (результати навчання)</p>	<p>Курс призначений для студентів, які люблять практичне програмування та вирішення завдань в області проєктування HW та IoT.</p> <p>Слухачі курсу отримають практичні навички:</p> <ul style="list-style-type: none"> - програмування на мові C, C++ для мікроконтролерів STM32, ESM32; - розвертання операційної системи реального часу RTOS на мікроконтролерах; - розробки програмного забезпечення для ОС RTOS на платі Global Logic Startet Kit STM32; - розвертання процесорних ядер за допомогою Toolchains. - розроблення програмного забезпечення в ОС Linux на платі та віртуальній машині; - тестування розроблених модулів ядра; - розроблення автотестів для вбудованих систем на мові Python; - використання системи контролю версій Git для роботи в команді; - генерації та аналізу мережного трафіку в інфраструктурі IoT; - оцінювати обчислювальні можливості вбудованих систем. <p>Незважаючи на поглиблену складову HW, студенти все ж отримають основні знання та навички з різних рівнів інфраструктури IoT. Як то організація та розробка системи кінцевих датчиків та сенсорів, мережного оточення та інтернет інфраструктури для</p>

	обміну даними між пристроями IoT та серверами обробки даних, архітектури Edge та хмарної інфраструктури аналітики даних – хмар IoT.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Після проходження курсу студенти зможуть розробити та тестувати програмно-технічні засоби обробки та пересилання інформації в комп'ютерних системах, що мають безпосередні зв'язки з реальними об'єктами.</p> <p>Отримані знання можна застосувати на практиці в будь якій сучасній IT компанії, особливо в компаніях, що займаються розробкою систем реального часу різного застосування:</p> <ul style="list-style-type: none"> – транспортні системи для автомобілів, метро, літаків, залізниць і кораблів; – контроль руху автомагістралей, повітряного простору, залізничних колій і судноплавних шляхів; – контроль технологічних процесів для електростанцій, заводів та споживчих товарів; – медичні системи для терапії, моніторингу пацієнтів; – військове використання; – виробничі системи з роботами; – телефонний, радіо- та супутниковий зв'язок; – комп'ютерні ігри, мультимедійні системи, текстові, графічні, аудіо- та відеоінтерфейси; – побутові системи для моніторингу та керування приладами; – контроль об'єктів, таких як тепло, світло, двері та ліфти; тощо.
Інформаційне забезпечення	Навчальна і робоча програма дисципліни, РСО, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	к.т.н., доцент Волокита А.М., асистент Каплунов А.В.

Дисципліна	Сучасні технології розробки програмного забезпечення
Рівень ВО	Перший (бакалаврський)
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з дисциплін: Інженерія програмного забезпечення, Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Архітектура комп'ютерів
Що буде вивчатися	<p>–основи термінології і основних понять. Основні методології розробки ПО. Методи вибору технологій для розробки. Огляд існуючих технологічних рішень які використовуються для створення проектів</p> <p>основи застосування теоретичних знань на прикладі розробки документації навчального проекту персоналізованого для кожного студента чи групи студентів за їх вибором</p>
Чому це цікаво/треба вивчати	–Існуючі потреби та тенденції на ринку праці показують що для успішної роботи програмістом потрібно не тільки вміти писати код(програмувати) також потрібне розуміння принципів розробки ПО в цілому. До 50% часу технічної співбесіди витрачається саме на

	питання які стосуються методологій розробки оскільки домінуючою методологією розробки ПО на даний час є AGILE отримати розуміння та практику розробки власного проекту з 0 дозволить достатньо підготуватися до майбутньої роботи
Чому можна навчитися (результати навчання)	Розібратися в реальному процесі розробки ПО. Підготуватися до того що буде очікувати студента після працевлаштування на реальне робоче місце.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розуміти як читати та писати проектну документацію для проекту вцілому чи його окремих компонент. Обирати методології та технології для реалізації проекту чи його компонент.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, навчально- методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 8 семестр
Викладач	ст. викладач Шевело О.П.

CATALOG
of selective academic disciplines
of the first (bachelor) level of higher
education for educational programs of
specialties:
121 Software engineering
123 Computer engineering

Instruction for catalog users

1. The number of selective subjects that a student can choose is determined by the curriculum. The curriculum also indicates the semester in which the selective discipline is taught, the form of semester control - assessment, types and volumes of educational classes, the amount (in ECTS credits) - 4 credits.
2. The direct choice of disciplines by students is carried out through willpower. Each student fills out a form in the "Electronic Campus" system, in which he indicates the disciplines he wishes to study in the next academic year (taking into account the number of disciplines and the study semester specified in the curriculum).
3. The catalog of selective disciplines is common to students of the 2nd and 3rd year of specialties 123 and 121 of the Department of Computer Engineering (FICE), so you need to use the information from the "Requirements before starting studies" section of the annotations and choose disciplines according to your level of training.
4. In case of impossibility of formation of educational groups of standard size for study of a specific selective discipline, students are given the opportunity to re-select within a specified period by joining the already formed educational groups.
5. At the request of a student who has chosen a certain elective discipline, he/she is allowed to join a group in which this Discipline is taught within the framework of another educational program.
6. A student cannot choose the same academic discipline twice.
7. If, for a valid reason, the student was unable to choose the disciplines on time, or discovered an error regarding his will, he applies to the dean's office with an application to enroll in the studies of his chosen disciplines, submitting the relevant documents. A student who has neglected his right to choose will be enrolled to study those disciplines that the head of the graduating department deems necessary to optimize study groups and streams.
8. The academic disciplines chosen by the student are indicated in his individual study plan.
9. More information on the procedure for students to exercise their right to freely choose academic disciplines can be found in the relevant Regulation on the procedure for students of higher education to exercise their right to freely choose academic disciplines.

CONTENT

Instruction for catalog users.....	112
Annotations of selective disciplines.....	114
AGILE software development methodology	114
Introduction to Data Science technology	115
Client software development technologies (Front-end).....	117
Server software development technologies (Back-end).....	118
C/Embedded programming technologies	120
Introduction to artificial intelligence	121
Fundamentals of software development on the Node.JS platform.....	122
Fundamentals of software development on the Java platform.....	122
Development of system programs	123
FPGA Programming Technologies	124
Program and data security systems	126
Computer Vision Technologies.....	127
Python programming technologies	130
Distributed information systems	131
Parallel and distributed computing	132
Hybrid computer systems	133
Management of IT infrastructure projects	134
Digital signal processing.....	136
Mathematical foundations of data protection and information security.....	136
Mobile computer networks	138
Basics of computer modeling	139
System programming in the Unix environment.....	140
Systems of automated design of computer systems	141
Testing and quality control (QA) of embedded systems	142
Methodology and organization of scientific research.....	144
Artificial intelligence methods and technologies.....	145
Basics of data science.....	146
Statistical methods of machine learning.....	147
Java programming language	149

Annotations of selective disciplines

Discipline	AGILE software development methodology
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Software engineering, Programming, Object-oriented programming, Algorithms and calculation methods, System programming, Algorithms and data structures, Architecture computers
What will be studied?	<ul style="list-style-type: none"> – the basics of terminology and basic concepts. Basic software development methodologies. Methods of selecting technologies for development. Overview of existing technological solutions used to create projects – the basics of applying theoretical knowledge on the example of developing documentation of an educational project personalized for each student or group of students of their choice
Why is it interesting/necessary to study?	<ul style="list-style-type: none"> – Existing needs and trends in the labor market show that for successful work as a programmer, one needs not only to be able to write code (program), but also to understand the principles of software development in general. Up to 50% of the time of a technical interview is spent precisely on issues related to development methodologies – since the dominant software development methodology is AGILE at the moment, gaining an understanding and practice of developing your own project from 0 will allow you to sufficiently prepare for future work
Why you can learn (results teaching)	Understand the real process of software development. Prepare for what will be expected of the student after employment in a real workplace.
How can you use acquired knowledge and skills?	Understand how to read and write project documentation for the project as a whole or its individual components. Choose methodologies and technologies for the implementation of the project or its component.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory work
Terminal control	Offset
Teacher	Sr. teacher Shevelo Oleksiy Pavlovich

Discipline	Introduction to Data Science technology
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	<p>Basic knowledge of:</p> <p>Data structures and algorithms, Algorithms and calculation methods, System programming, System software, Organization of databases, Introduction to the Linux OS, Programming, Computer architecture, Computer networks, Discrete mathematics, Computer logic, Software engineering</p>
What will be studied?	<p>The purpose of the discipline is to provide a comprehensive, thorough theoretical basis and powerful practical skills in software implementation of methods, mathematical models and algorithms of Data Science technological processes (data research): data - information - knowledge - knowledge manipulation - visualization.</p> <p>The specificity of the course consists in considering, along with classic Data Science methodologies, advanced author's developments obtained during the implementation of practical R&D projects.</p> <p>The theoretical foundations of Data Science are provided in the form of lectures with a mandatory demonstration of the considered algorithms in the form of examples of program code.</p> <p>Practical skills in the application of Data Science technologies are acquired in laboratory classes. At the same time, special attention is paid to software engineering processes.</p> <p>The practical part of the discipline is focused on the application of the high-level Python programming language with the study of the functionality of the libraries: Pandas, NumPy, Matplotlib, scikit-learn.</p> <p>The discipline reveals the essence of Data Science technological processes: data processing for the purpose of obtaining information - information processing for the purpose of discovering knowledge - using skills in practice - visualization of results.</p> <p>The discipline consists of two interrelated units:</p> <ol style="list-style-type: none"> 1. Methodological foundations of Data Science: <ol style="list-style-type: none"> 1.1. Applied statistical data analysis (data models; statistical analysis of experimental sample characteristics; processing of anomalous measurements; evaluation, extrapolation and interpolation by trend models - recurrent smoothing and smoothing of the accumulated sample; construction of nonlinear models of experimental data using differential transformations); 1.2. Multi-criteria decision-making methods (multi-criteria

	<p>evaluation; multi-criteria identification; multi-criteria distribution of resources; multi-criteria structural and parametric synthesis of systems);</p> <p>1.3. Intelligent data analysis. (Technologies: OLAP, Data Mining, Text Mining, Image Mining, Knowledge discovery, Speech and language recognition);</p> <p>1.4. Application of artificial intelligence for Data Science technologies (artificial neural networks; multi-criteria optimization neural networks; model methods and algorithms of self-organization and situational analysis).</p> <p>2. Technological aspects of Data Science:</p> <p>2.1. Algorithms and technologies for prediction the dynamics of changes in financial and stock markets (according to statistical and alternative models);</p> <p>2.2. Algorithms and technologies for determining credit risks for banking CRM systems (scoring and multi-criteria models);</p> <p>2.3. Algorithms for identifying current situations for production CRM systems and critical infrastructure facilities (multifactor analysis and Computer Vision technologies).</p>
Why is it interesting/necessary to study?	Mastering the knowledge, skills and abilities required for the positions: Data Scientist; Data Engineer; Data Analyst; Risk Team, etc.
What will be learned?	<p>1. Knowledge and skills of application and implementation of basic Data Science algorithms:</p> <ul style="list-style-type: none"> – applied statistical data analysis; – multi-criteria decision-making methods; – intelligent data analysis; – application of artificial intelligence for Data Science technologies; – visualization of Data Science results. <p>2. Knowledge and skills of using basic Python libraries to implement Data Science algorithms: Pandas, NumPy, Matplotlib, scikitlearn.</p>
How can you use acquired knowledge and skills?	<p>Acquired competencies in Data Science can be applied to projects in the following applied industries and technologies:</p> <ol style="list-style-type: none"> 1. Data analysis for e-commerce tasks; 2. Data analysis for industrial and infrastructure CRM systems; 3. Analysis of visual and geospatial data of various directions; 4. Data analysis for medical information systems.
Information support	Educational and working programs of the discipline, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works
Semester control	Offset
Teacher	Doctor of Science, professor Pisarchuk Oleksii Oleksandrovich

Discipline	Client software development technologies (Front-end)
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	<p>Basic knowledge of:</p> <p>Algorithms and calculation methods, System programming, System software, Algorithms and Data structures, Organization of databases, Introduction to the Linux OS, Programming, Computer architecture, Computer networks, Discrete mathematics, Computer logic, Software engineering, Computer systems</p>
What will be studied?	<p>The purpose of the discipline is to provide thorough theoretical knowledge and specific practical skills in the development of Front-end (client) software.</p> <p>The educational discipline is aimed at the formation of competencies in the development of the Front-end part of applied Web-applications. In combination with the educational discipline «Technologies for the development of server software (Back-end)» (7th semester), the acquisition of comprehensive knowledge and practical skills in the design and development of application software, based on modern and promising client-server technologies of distributed processing and information exchange, is ensured.</p> <p>The theoretical and practical components of the educational discipline are aimed at acquiring competencies during the implementation of a specific project on the development of the Front-end part with simplified implementation of the Back-end component.</p>
Why is it interesting/necessary to study?	Mastering the knowledge, skills and abilities required for the positions: Front-end developer, Front-end Engineer, Front-end Team Lead, DevOps engineer, etc.
What will be learned?	<ol style="list-style-type: none"> 1. Knowledge and skills in the application and implementation of client service technologies to create effective front-end parts of modern Web applications: <ul style="list-style-type: none"> - front-end interface prototyping; - user interface programming technologies. 2. Knowledge and skills of using basic Front end technologies: HTML5; CSS3; JavaScript; TypeScript; jQuery; React and Flux architecture; Angular 6 and MVC architecture.
How can you use acquired knowledge and skills?	<p>Acquired competences in the development of Front-end software can be applied to projects of the following applied industries and technologies:</p> <ol style="list-style-type: none"> 1. E-commerce; 2. Electronic document management;

	3. CRM systems of various applied directions, built on technologies of distributed processes of collection, storage, processing and exchange of information.
Information support	Educational and working programs of the discipline, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works
Semester control	Offset
Teacher	Doctor of Science, professor Pisarchuk Oleksii Oleksandrovich

Discipline	Server software development technologies (Back-end)
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, Object-Oriented Programming, Algorithms and calculation methods, System programming, Data structures and algorithms, Discrete mathematics, Software engineering.
What will be studied?	<p>The purpose of the discipline is to provide thorough theoretical knowledge and specific practical skills in server software development.</p> <p>The educational discipline is aimed at the formation of competencies in the development of the Back-end part of applied Web-applications. In combination with the educational discipline «Technologies of programming user interfaces (Front End)» (8th semester), the acquisition of comprehensive knowledge and practical skills in the design and development of application software based on modern and promising client-server technologies of distributed processing and information exchange is ensured.</p> <p>The theoretical and practical components of the educational discipline are aimed at acquiring competencies during the implementation of a specific project on the development of the Back-end part with a simplified implementation of the Front-end component.</p> <p>The discipline consists of the following blocks:</p> <p>3. Methodology and technologies of building and creating client-server (Web) applications (model «client-server», cloud services, network level, exchange protocol and Internet standards: TCP/IP, WWW, XML, JSON, HTTP, HTTPS, FTP, Telnet, MIME; URL and URI; the main tools of errors research (ICMP, ping, traceroute);</p>

	<p>sockets, IP and port's addressing, using proxy-servers; name search services: DNS, whois; remote access services: Telnet, SSH, Remote Desktop, VNC);</p> <p>4. API and its protocols. Organization of the access restriction and authorization system (rules). Frameworks and toolset for Back-end development. Interaction with the database by using the object manager (ORM – Hibernate, Doctrine, typeORM, Eloquent). Technologies of back-end software development: PHP (Symfony, Laravel), Java (Spring Boot, Java EE), JavaScript (Express, NestJS);</p> <p>5. Specification of web-services. Code testing (OPENAPI3 documentation; creating of schemes; basics of Test Driven Development – unit, integration and e2e tests (PHPUnit, JUnit, Jest)); catching and handling errors.</p> <p>6. Introduction to DevOps. Component parts of DevOps: Development (Software Engineering), Operations, Quality; Assurance; configuration of services Apache, PHP, MySQL; DevOps integration into the development process of software; DevOps and Digitalization; CI/CD development; High Availability.</p>
Why is it interesting/necessary to study?	Mastering the knowledge, skills and abilities required for the positions: Back-end developer; Back-end Engineer; Back-end Team Lead; DevOps Engineer, etc.
What will be learned?	<ol style="list-style-type: none"> 1. Knowledge and skills of client-server application and implementation technologies for creating effective Back-end parts of modern Web applications: <ul style="list-style-type: none"> – methodology and technologies of construction and creation of Back-end parts client-server applications; – modern server software development technologies; – code testing; – basics of implementing DevOps tasks. 2. Knowledge and skills of using basic Back-end approaches: frameworks, testing, API development.
How can you use acquired knowledge and skills?	<p>Acquired competences in the development of server software (Back-end) can be applied to projects of the following applied industries and technologies:</p> <ol style="list-style-type: none"> 1. E-commerce; 2. Electronic document management; 3. CRM systems of various applied directions, built on technologies of distributed processes of collection, storage, processing and exchange of information.
Information support	Educational and working programs of the discipline, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works
Semester control	Offset
Teacher	Doctor of Science, professor Pisarchuk Oleksii Oleksandrovich

Discipline	C/Embedded programming technologies
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Discrete mathematics, Computer logic, Computer arithmetic, Programming, Computer architecture, Introduction to the Linux OS.
What will be studied?	Creating software for ARM-based microcontrollers. Implementation of software configuration management and development of embedded programs. Deploying a processor core using Toolchain. Using the Git version control system. Software development in Linux on a virtual machine. Development of a project with its own firmware based on hardware for programming and debugging microcontrollers. Using BeagleBone Black, Raspberry Pi, STM StarterKit GlobalLogic platforms.
Why is it interesting/necessary to study?	Embedded systems and IoT devices require software to function and achieve their operational purpose. However, writing this software from scratch for each type of hardware is not practical. The most common approach is the use of operating systems to solve the main task of managing hardware resources and providing services for user applications. Field of application. Development of solutions for the Internet of Things (IoT), embedded systems, smart systems. Development of hardware systems for the automotive industry, industrial automation, high-performance computing, artificial intelligence and machine learning.
What will be learned?	<ul style="list-style-type: none"> – Install and configure the Linux operating system (Linux kernel, U-Boot, BusyBox) for the architecture of ARM processors by compiling the source code. – Program in the Bash shell. – Use the Git version control system – To program in C for Linux OS for processors of the ARM architecture – Program your own modules for the processor core – Perform testing of own product on the basis of modern platforms for development. – Work in a team.
How can you use acquired knowledge and skills?	<p>The knowledge gained is sufficient to design the architecture and implement your own smart device, debug it and configure it to work in an IoT environment.</p> <p>The experience and practice gained are sufficient to continue independent professional training in the field of developing new IoT systems using any hardware and software available on the market.</p> <p>The acquired experience, knowledge and practice are sufficient to successfully pass interviews in IT companies engaged in the development of embedded systems and IoT.</p>
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, practical and seminar classes, laboratory works
Semester control	Offset
Teacher	Assistant Kaplunov Artem Volodymyrovych

Discipline	Introduction to artificial intelligence
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian, English
Department	Computer Engineering
Prerequisites	Basic knowledge of: Algorithms and calculation methods, System programming, System software, Algorithms and Data structures, Organization of databases, Programming, Computer architecture, Computer networks, Discrete mathematics, Software engineering, Parallel programming
What will be studied?	Basic knowledge about the context, terms, problems and current state of research in the field of artificial intelligence. Overview of the main principles and approaches to the study of artificial intelligence systems. An overview of the main areas of research and relevant artificial intelligence tools: genetic algorithms, evolutionary methods, machine learning, deep learning, reinforcement learning, etc. Overview of the main models and algorithms with examples of their application.
Why is it interesting/necessary to study?	In recent years, the field of artificial intelligence has experienced a period of real uplift due to the rapid development of new models of neural networks, software tools and the rapid development of computing resources based on graphics (GPU) and tensor (TPU) accelerators. Research on artificial intelligence methods is currently showing the fastest growth rates in scientific research and wide practical application in many areas of life around us: from personal assistants in smartphones to self-driving cars.
What will be learned?	Theoretical knowledge and basic practical experience in the application of various methods of artificial intelligence to existing practical problems in a wide range of applications.
How can you use acquired knowledge and skills?	The acquired knowledge will allow to understand the place and role of artificial intelligence methods in the general context of information technologies. This discipline is a necessary stage for preparing for the professional level of mastering specialized knowledge and skills in the field of artificial intelligence, which will be taught in the following educational disciplines, which are devoted to a more detailed study of individual methods of artificial intelligence.
Information support	Educational and working programs of the discipline, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works
Semester control	Offset
Teacher	Doctor of Science, professor Hordienko Yury Hryhorovych

Discipline	Fundamentals of software development on the Node.JS platform
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, Object-Oriented Programming, Algorithms and Data structures, Computer networks.
What will be studied?	Development of programs in the JavaScript language. Studying of basic principles, basic constructions of language. Using the Node.Js platform to develop web systems. Development tools for server components and front-end applications.
Why is it interesting/necessary to study?	Mastering modern technologies for the development of web systems for the creation of multi-platform software.
What will be learned?	Create effective web systems software.
How can you use acquired knowledge and skills?	Ability to create software for a variety of OSs and hardware platforms. Ability to create application software for computer systems and networks.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works, independent work
Semester control	Offset
Teacher	Sr. teacher Shemsedinov Timur Gafarovych

Discipline	Fundamentals of software development on the Java platform
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, Object-Oriented Programming, Algorithms and Data structures, Computer Discrete mathematics.

What will be studied?	Principles of construction of the Java language, basic structures, libraries. Implementation of the principles of object-oriented programming, modularity, working in the network and with databases. Classes, interfaces, packages and other basic elements of building programs. Program development tools, integrated software development and debugging environments. Java Virtual Machine. Organization of multithreaded execution.
Why is it interesting/necessary to study?	Mastering modern Java technologies for creating platform-independent software, technologies for developing complex information systems for solving a wide range of tasks.
What will be learned?	Create effective software based on Java technologies, learn and master modern software development tools based on an object-oriented approach and popular design and programming patterns.
How can you use acquired knowledge and skills?	Ability to build software for a variety of operating systems and hardware (network, mobile, embedded) platforms. Ability to create application software for computer systems and networks.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works, independent work
Semester control	Offset
Teacher	Sr. teacher Aleshchenko Oleksiy Vadimovych

Discipline	Development of system programs
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, Object-oriented programming, Algorithms and calculation methods, System programming, Algorithms and Data structures, Introduction to the Linux OS
What will be studied?	<ul style="list-style-type: none"> – composition of system software of modern computer systems; – technology of step-by-step development of complex software products Incremental Approach on the example of developing system programs; <ul style="list-style-type: none"> – practical application of this technology during the development of system programs; – offsetting of developed system programs.

Why is it interesting/necessary to study?	<ul style="list-style-type: none"> – system software is present in any computing device, so the demand for those who understand it – will always be high; – system software is the basis of the functioning of computer systems, it is it that dictates the requirements for application programs, therefore its developers will always be ahead of programmers applicants; – the development of system software is usually carried out by the most well-known companies that determine the trend of computer technology development for many years to come.
What will be learned?	<ul style="list-style-type: none"> – develop and Offset system programs, understand messages operating systems, perform code optimization.
How to use acquired knowledge and skills (competence)	<ul style="list-style-type: none"> – in the development and improvement of operating systems, specialized software systems, computer security systems, programming systems, individual components of system software: drivers, file management programs, etc.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory work, independent work.
Semester control	Offset
Teacher	Ph.D., associate professor Pavlov Valery Georgiyovych

Discipline	FPGA Programming Technologies
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of computer logic, computer arithmetic, discrete mathematics. Basic knowledge of computer architecture, including knowledge of the general structural and functional organization of Von-Neumann architecture computers, general principles of firmware management, processor core structure, and general principles of firmware level organization. Skills programming.

What will be studied?	Digital design and engineering methods for FPGAs. Hardware description languages for the development of digital devices. Design of digital devices using Verilog. Functional modeling and structural synthesis in modern CAD. Practical tasks are initially based on simple examples of programming computer hardware components. Next, an introduction to the main stages of the full cycle of development of IP-cores of digital technology is offered, including functional synthesis, structural synthesis, verification of developments and programming of the FPGA microcircuit. The basic issues of creating OffsetBench levels in the Verilog language are considered. CAD models ModelSym MentorGraphic, Quartus II are used to check the correctness of the created projects and debugging. Issues of integration of CAD Quartus II and ModelSim are considered. Altera DE2 Board (Cyclon II), DE10 Standard Board is used for experiments (Cyclon IV), DE1 SoC Board (Cyclon V).
Why is it interesting/necessary to study?	Programmable logic (FPGA) has recently become the main technology used to create electronic systems in various fields of application. The acquired skills and knowledge will be relevant in the areas of developing solutions for the Internet of Things (IoT), embedded systems, smart systems. Development of hardware systems for the automotive industry, industrial automation, high-performance computing, machine artificial intelligence teaching.
What will be learned?	Design digital circuits and devices using modern design methods and hardware design languages. Gain programming skills in Verilog. Use modern professional tools for digital design and modeling. Perform functional modeling and debugging of devices in ModelSym CAD. Perform structural synthesis in CAD Quartus II Altera. Implement devices on modern Altera DE2 boards Board, DE10 Board, DE1 SoC Board.
How to use acquired knowledge and skills (competence)	Basic knowledge in the use of the hardware design language, Verilog, as well as experience and skills in the use of modern CAD and Altera products is sufficient for continuing professional training in the direction of hardware design for complex systems on FPGA and ASIC, system-on-chip (SoC), embedded systems, smart systems. software and hardware implementations of artificial intelligence systems. The acquired basic knowledge of programming and digital design methods will allow, if necessary, to quickly switch to the use of hardware programming language VHDL, chips of other FPGA manufacturers (Xilinx), as well as other CAD for functional and structural synthesis on FPGA and ASIC (Synopsys, Aldec, Cadence). The acquired basic knowledge will help in further professional work learning, including independently.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, practical and seminar classes, laboratory works
Semester control	Offset

Teacher	Doctor of Science, associate professor, professor Klymenko Iryna Anatoliivna
----------------	--

Discipline	Program and data security systems
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Computer architecture, Computer networks, Algorithms and Data structures, Programming, Discrete mathematics, Probability theory and Mathematical statistics.
What will be studied?	This course is devoted to an in-depth study of specific issues related to the mathematical and algorithmic aspects of information protection in computer systems and the design and development of secure software and security systems. Methods and algorithms of protection during data transmission and storage, development of software and hardware means of protection are studied. In particular, protected network protocols, encryption methods, cryptography, and steganography. The issue of creating and organizing secure services using cloud technologies in various industries, in particular, e-commerce, e-banking, functioning of cryptocurrencies.
Why is it interesting/necessary to study?	The modern realities of functioning of the digital society require using and develop computer tools and systems resistant to unauthorized access and use of programs and data. Every qualified specialist related to software must know the principles of information security and methods of protecting programs and data and be able to apply them competently.
What will be learned?	Learn to make decisions about the choice of data protection structure; understand mathematical methods and algorithms of cryptography and data encryption. Develop information protection software that is resistant to hacker attacks. Organize secure storage, processing and transmission of information using technologies of distributed and cloud systems.
How can you use acquired knowledge and skills?	The ability to analyze, select, develop and apply methods and tools to ensure information security, cybersecurity. The ability to perform work as an analyst of computer systems and software. Being a qualified software developer and user of hardware and software complexes.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works, independent work
Semester control	Offset

Teacher	Ph.D., associate professor Volokita Artem Mykolayovych
----------------	--

Discipline	Computer Vision Technologies
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Algorithms and calculation methods, System programming, System software, Algorithms and Data structures, Organization of databases, Programming, Computer architecture, Computer networks, Discrete mathematics, Computer logic, Software engineering, Parallel programming
What will be studied?	<p>The purpose of the discipline is to provide a thorough theoretical basis and powerful practical skills in software implementation of methods, mathematical models and algorithms of Computer Vision technologies.</p> <p>The theoretical foundations of Computer Vision are provided in the form of lectures with a mandatory demonstration of the practical implementation of the considered algorithms in the form of program code examples.</p> <p>Practical skills in the application of Computer Vision technologies are acquired in laboratory classes, which are built on the principle of increasing the functionality of developed scripts. At the same time, special attention is paid to software engineering processes.</p> <p>The practical part of the discipline is focused on the application of the high-level Python programming language with the study of the capabilities of the graphics libraries Graphics, Tkinter, Matplotlib, NumPy (for the "raw" implementation of Computer Vision algorithms) and specialized packages such as PIL/Pillow, OpenGL, OpenCV for creating software modules of a completed practical orientation.</p> <p>The discipline reveals the essence of the stages of the classic pipeline of processing digital graphic images: image synthesis (spatial transformation) - rasterization (realistic image, digital processing) - vectorization (object identification and target image processing). This is implemented in the following topics:</p> <ol style="list-style-type: none"> 1. Digital image synthesis (2D, 3D - objects, graphic method, analytical and non-analytical methods); 2. Basic geometric transformations of 2D, 3D objects (scaling, moving, rotating, projections); 3. Rasterization and raster images, creation principles, characteristics and processing algorithms (rasterization, brightness change, filtering);

	<ol style="list-style-type: none"> 4. Vectorization and vector images, mathematical models, approximation, interpolation and smoothing algorithms (MNC, Splines, Bezier curves, etc.); 5. Realistic image models, augmented reality, light and shadow models: 3D pipeline, rendering; 6. Digital image processing: filtering; color correction; segmentation; vectorization; morphological processing; recognition; 7. Methodological, technological and applied aspects of Computer Vision: sensor data processing; navigation; Artificial Intelligence; management.
Why is it interesting/necessary to study?	Mastering the knowledge, skills and abilities required for the positions: Software Developer with Computer Vision; Embedded Developer for Computer Vision systems; Computer Vision Research Engineer, etc.
What will be learned?	<ol style="list-style-type: none"> 1. Knowledge and skills of application and implementation of basic Computer Vision algorithms: <ul style="list-style-type: none"> – synthesis of 2D, 3D graphic objects and their geometric transformations; – synthesis and processing of raster images; – synthesis and processing of vector images; – formation of realistic images; – digital image processing; – applied aspects of Computer Vision; – comprehensive implementation of stages of the digital image processing pipeline for Computer Vision tasks and other applied fields. 2. Knowledge and skills of using basic Python libraries to implement Computer Vision algorithms: Matplotlib, NumPy PIL/Pillow, OpenCV.
How can you use acquired knowledge and skills?	<p>Acquired competencies in Data Science can be applied to projects in the following applied industries and technologies:</p> <ol style="list-style-type: none"> 1. 3D scene reconstruction and navigation in it (Structure-from-Motion, Road Scene Understanding and Autonomous Driving); 2. Scene segmentation and plot understanding (Significance maps, video and 3D segmentation, 3D stream, multiple tracking, object detection, activity detection and prediction, group analysis, object detection and recognition); 3. Image recognition, object identification (Convolution networks, Deep learning, Image Retrieval, Object Detection).
Information support	Educational and working programs of the discipline, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works
Semester control	Offset

Teacher	Doctor of Science, professor Pisarchuk Oleksii Oleksandrovich
----------------	---

Discipline	Python programming technologies
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, Object-Oriented Programming, Algorithms and calculation methods, System programming, Algorithms and Data structures, Software engineering, Basics of parallel programming
What will be studied?	Methods of converting project documentation into a real code. Methods of process optimization for writing code to achieve tasks according to trade-off strategy, working with risks, etc. Skills of allocating tasks in time and planning a full development cycle. Code testing. Code deployments on server infrastructure.
Why is it interesting/necessary to study?	One of the major issues of modern software is the need not only to write high-quality code, but also the ability to meet deadlines without significant quality losses. It is also necessary to be able to work with risks, priorities, etc.
What will be learned?	The acquired knowledge will allow you to learn how to transform a project documentation into a real code, meeting the requirements both in terms of time and code quality. Working in the development team.
How can you use acquired knowledge and skills?	To develop software in a team or independently.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, practical and seminar classes, laboratory works
Semester control	Offset
Teacher	Sr. teacher Shevelo Oleksiy Pavlovich

Discipline	Distributed information systems
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Software engineering, Programming, Object-oriented programming, Algorithms and calculation methods, System programming, Algorithms and data structures, Introduction to the operating room Linux systems, Computer architecture, Discrete mathematics,
What will be studied?	<ul style="list-style-type: none"> – architectural templates for the implementation of distributed information systems, the basics of service-oriented architecture, principles and protocols of interaction of web services, principles of building software interfaces, ways of integrating distributed software resources; – basics of software interaction with SQL and NoSQL databases, concepts of streaming data processing; – the basics of technological equipment for the process of designing, documenting, developing and Offsetting software; – basics of building web applications (Single Page Application, SPA, Power Web Application, PWA) using modern software tools (React, Angular, Vue); – ways of implementing the process of collecting and placing software systems using PAAS cloud services.
Why is it interesting/necessary to study?	Today, the possession of knowledge, abilities and skills related to the process of designing distributed systems of collective use, the process of developing appropriate software and organizing technological support for all stages of the software life cycle is one of the key competitive advantages in the IT labor market.
What will be learned?	Carry out the design of information systems based on the use of modern architectural principles with the use of modeling and documentation tools at all stages of the software life cycle, develop information and software for such systems and organize technological support the process of their development and placement.
How to use acquired knowledge and skills (competencies)?	Carry out design and development of modern information systems for collective use, organize technological support for the software development process, use tools for organizing distributed data processing, use relational, document-oriented and graph databases, integrate own software with publicly available web services, place software in PAAS cloud environment.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory work
Terminal control	Offset
Teacher	Ph.D., associate professor Boldak Andriy Oleksandrovich

Discipline	Parallel and distributed computing
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, Object-oriented programming, Algorithms and calculation methods, System programming, Algorithms and data structures, Computer architecture, Analytic geometry, Engineering Software,
What will be studied?	<p>Methods of organizing calculations in parallel computer systems with different structural organization: systems with shared memory, systems with local memory, distributed systems. The life cycle of program development for a parallel system, which includes:</p> <ul style="list-style-type: none"> – development of a parallel calculation algorithm. – development of flow execution algorithms with a solution to the task of mutual exclusion and synchronization. – development of a structural diagram of flow interaction for selected parallel programming languages (libraries) and means of flow organization: semaphores, mutexes, events, critical sections, – methods of debugging and Offsetting a parallel program – research of the percentage of loading of processors (cores) of the system, determination of execution time, calculation of the acceleration factor (speedup). <p>Modern parallel programming languages and libraries are used in laboratory works: Java, C#, WinAPI, Ada, OpenMP, MPI, TBB.</p>
Why is it interesting/necessary to study?	Modern computers (from tablets to supercomputers) are equipped with multi-core processors, so creating efficient software for them requires the use of threads for optimal use of multi-core hardware. Multi-core processors are also part of most gadgets, such as smartphones, mp3 players, video players, game consoles, which allows you to speed up (improve) sound processing and image.
What will be learned?	Ability to develop software for parallel and distributed computer systems of various purposes, analyze tasks for the presence of parallelism, build optimal parallel algorithms, use threads to build a program, organize optimal interaction of threads depending on the structure of the computer system, debug a parallel program, prevent and get rid of deadlocks. Use of modern parallel programming languages and libraries.
How to use acquired knowledge and skills (competencies)?	The acquired basic knowledge will help in further professional training, including independently, expand the perspective of career opportunities in the field of software development for any parallel computer systems, real-time systems, and as well as creation of server-client applications.

Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, practical and seminar classes, laboratory works
Semester control	Offset
Teacher	Ph.D., associate professor Korochkin Oleksandr Volodymyrovych

Discipline	Hybrid computer systems
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Probability theory and mathematical statistics, System programming. System software, Computer systems, Computer architecture
What will be studied?	Fundamentals of designing analog and hybrid operating units, analog-to-digital and digital-to-analog information converters, analog and hybrid processors, methods of mathematical modeling using operating units, methods of preparing problems for analog and hybrid systems, programming methods for analog and hybrid computer systems
Why is it interesting/necessary to study?	The acquired knowledge allows mastering the basic methods of building specialized computer systems
What will be learned?	Using of modern technical and software design tools, analog and hybrid computer systems
How can you use acquired knowledge and skills?	Hybrid CSs will allow to effectively solve the following tasks: <ul style="list-style-type: none"> – simulation of automatic control systems in real time that contain analog and digital devices; – real-time reproduction of processes containing high-frequency components and variables in a wide amplitude and frequency range; – statistical modeling; – modeling of biological systems; – solutions of equations in partial derivatives;

	– optimization of management systems.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works
Semester control	Offset
Teacher	Ph.D., associate professor Selivanov Viktor Levovych

Discipline	Management of IT infrastructure projects
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, Computer architecture, Discrete mathematics, Computer logic, Software engineering, Basics of parallel programming, Object-Oriented Programming, Algorithms and calculation methods, System programming, Algorithms and Data structures, Computer networks
What will be studied?	<p>The main goal of the discipline is the acquisition of theoretical knowledge and practical skills of project management and teamwork: gaining of soft skills by applying one's hard skills to solving practice problems. It is implemented in the form of organization and implementation of a specific startup project in the field of information technology and computer engineering. The content of the discipline takes into account the best practices of leading IT enterprises and involves the involvement of mentors - practitioners. The result of working out the tasks of the discipline is a completed practical project.</p> <p>The discipline involves the study of two interrelated components.</p> <p>I. Project management:</p> <p>Organization and implementation of the pre-project R&D stage; development technical proposal and technical task; requirements engineering (business analysis); architectural design; system development (software); quality assessment (verification and testing); implementation and basics of DevOps; accompaniment; organization of processes (life cycle models) according to classic and Agile models; reporting and submission of the project.</p> <p>II. Group dynamics:</p> <p>Theory and practice: professions and positions in IT enterprises and</p>

	<p>their scope of responsibilities; drafting an effective resume; conducting an interview; selection of talented personnel; formation of groups and teams, organization of teamwork; leadership; conflictology; there-management; preparation, organization and conducting of effective meetings, presentations, negotiations.</p>
Why is it interesting/necessary to study?	<p>The acquired knowledge and skills ensure understanding, participation and implementation of modern processes of team development of complex hardware and software systems and complexes at the level of advanced national and world practices.</p>
What will be learned?	<p>Knowledge and skills of organization, application and implementation:</p> <ul style="list-style-type: none"> – team executing of projects using all phases of the life cycle (from gathering requirements to implementation); – execution of projects using different development methodologies (classical and Agile) and in different roles (developer, tester, analyst, DevOps, manager, etc.); – using of automated tools for team development of projects; – business communications (meetings, presentations, active listening, written communication, interviewing, etc.); – teamwork (giving feedback, sharing experiences, resolving conflict situations, etc.); – time management; – practical solutions to problem situations; – search, collection and processing of information necessary for the implementation of practical tasks.
How can you use acquired knowledge and skills?	<p>Team work on complex technical projects in accordance with modern requirements for processes and technologies for their implementation.</p>
Information support	<p>Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.</p>
Form of conduct classes	<p>Lectures, laboratory works</p>
Semester control	<p>Offset</p>
Teacher	<p>Doctor of Science, professor Pisarchuk Oleksii Oleksandrovich</p>

Discipline	Digital signal processing
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Higher mathematics, Programming, Discrete mathematics, Computer architecture, Computer systems
What will be studied?	Mathematical foundations and algorithms of digital signal processing (DSC) and images, as well as methods of their assembly, programming and hardware implementation.
Why is it interesting/necessary to study?	Modern hardware and software tools that are common in the IT industry are very often based on the implementation of COS algorithms. These are, for example, means of automation, telecommunications, image processing, computer games, pattern recognition and artificial intelligence, IoT. Without knowledge of TOS algorithms, it is almost impossible to Offset and modernize and develop such tools.
What will be learned?	<ul style="list-style-type: none"> – to choose, model and modernize the algorithms of the TSO; – to develop applications that execute TSO algorithms.
How to use acquired knowledge and skills (competencies)?	<ul style="list-style-type: none"> – to carry out analysis, diagnostics, selection of existing COS means, – carry out the modernization of existing and development of new means of COS, – to implement COS algorithms in IoT and artificial intelligence tools. Knowing the basics of COS and having experience in its programming is a mandatory requirement for employment in most companies involved in the development and distribution of applications and automation equipment. – telecommunications, artificial intelligence, IoT.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory work
Semester control	Offset
Teacher	Doctor of Science, professor Sergienko Anatoly Mykhailovych

Discipline	Mathematical foundations of data protection and information security
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Computer architecture, Computer systems and networks, Algorithms and Data structures, Discrete mathematics, Probability

	theory and Mathematical statistics, System programming
What will be studied?	Mathematical foundations of modern cryptography: irreversible transformations of Boolean algebra, number theory, elliptic curves and finite Galois fields. Methods of synthesis of cryptographic Boolean transformations, construction of data protection algorithms based on them. Modern algorithms and protocols of information protection based on Boolean functions. Attacks on algorithms and protocols of this class. Methods of building cryptographic protection systems based on irreversible transformations of number theory. Algorithms and protocols of information protection based on irreversible transformations of number theory and methods of breaking them. Mathematical foundations of the organization of homomorphic encryption of data during their processing in the clouds. Methods of breaking data protection tools by means of temporal or static analysis of power consumption dynamics. Mathematical principles of software polymorphism in the protected organization of calculations. Quantum computing and quantum cryptography.
Why is it interesting/necessary to study?	The progressive development of information integration will require qualitatively new means of data protection and demarcation of access to them in integrated environments. Creating and using such tools is impossible without mastering the fundamentals of cryptography. Professional analysis of the real level of security of data and programs requires special mathematical training.
What will be learned?	To gain thorough knowledge in special sections of mathematics, which are the basis of modern and promising means of cryptographic data protection. Master the skills of creating basic blocks of cryptographic data protection, building algorithms and protocols based on them. Acquire the ability to analyze the level of protection of information security tools. Know the protocols of cryptographic protection of information in computer systems and networks.
How can you use acquired knowledge and skills?	Develop, improve and adapt software and hardware tools for cryptographic data protection to the conditions of specific use. Create homomorphic ciphers to protect data directly in the process of their remote processing in the clouds, depending on its characteristics. To carry out a professional analysis of information security of data in view of the ability to resist various types of attacks.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works, independent work
Semester control	Offset
Teacher	Ph.D., associate professor Markovskiy Oleksandr Petrovich

Discipline	Mobile computer networks
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Computer networks, System programming, Computer architecture
What will be studied?	Basic construction concepts, architecture, communication system of mobile computer networks, wireless broadband networks, wireless personal networks, wireless 5G networks, software-configured mobile networks, mobile cloud technologies, mobile Internet, basics of security of mobile computer networks.
Why is it interesting/necessary to study?	The course is aimed at the formation of theoretical knowledge and practical skills in the building, management, modernization, monitoring of modern mobile computer networks.
What will be learned?	To learn the principles of formation of theoretical knowledge and practical skills in construction, operation, monitoring and performance analysis of modern mobile computer networks.
How can you use acquired knowledge and skills?	The ability to select the type, plan and implement mobile computer networks, manage network resources, select a complex of necessary hardware and software tools for a mobile computer network, expand and modernize networks, monitor and analyze performance, diagnose and solve problems.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works
Semester control	Offset
Teacher	Doctor of Science, professor Kulakov Yury Oleksiyovych

Discipline	Basics of computer modeling
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, Object-oriented programming, Algorithms and computational methods, System programming, Algorithms and data structures, Introduction to the Linux operating system, Discrete mathematics, Software engineering, Theory probabilities
What will be studied?	<ul style="list-style-type: none"> – Imitation, mathematical, logical, evolutionary, structural-linguistic modeling methods. Fundamentals of analog and physical modeling. Modeling methodology as a technology of scientific research and design of computer systems and their software. - Methods of creating models taking into account the requirements for their adequacy, methods and techniques of their software implementation and use in practice, methods of evaluating the results obtained with the help of models. Using models to optimize development and assess its reliability. Methods of evaluating the adequacy of models. – The most common models of computer systems in practice are Petri nets and modeling software packages. - Mathematical models of process optimization and methods of their software implementation. Dynamic and integer programming. Methods of evolutionary modeling based on genetic algorithms and technologies artificial intelligence.
Why is it interesting/necessary to study?	<ul style="list-style-type: none"> – acquired knowledge and practical skills make it possible to scientifically and methodically competently apply modeling to solve a wide range of practical problems of designing systems, software, organizing data transmission in local and global networks, evaluating the reliability of programs and the level of information security. – to find optimal solutions for project solutions and natural-scientific and economic problems with the help of modeling. – to analyze the behavior of systems and programs in various, including including critical situations, as well as in the presence of errors during their development.
What will be learned?	<ul style="list-style-type: none"> – Analyze the possibilities of solving a wide range of practical problems using modeling methods; make a justified choice of the model and its construction taking into account the requirements for its adequacy, competently assess the reliability of the obtained modeling results; – to master practical ways and techniques of programming simulation, mathematical and evolutionary models – Use mathematical models of process optimization and dynamic programming to create effective programs. – To evaluate the quality parameters of systems and programs, in

	<p>particular, the reliability of their operation in various applications, using computer models</p> <ul style="list-style-type: none"> – situations, as well as the level of data and program security
How to use acquired knowledge and skills (competencies)?	<ul style="list-style-type: none"> –To use the arsenal of computer modeling methods as a tool of scientific research in technical, natural-scientific, medical, economic and social spheres. –Apply computer modeling methods and existing software products for effective design of systems and programs taking into account the specified requirements for their quality characteristics –Optimizing project solutions using computer modeling; –Analyze the behavior of systems and programs in critical situations situations, evaluate their reliability, search for errors.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory work
Semester control	Offset
Teacher	Ph.D., associate professor Volokita Artem Mykolayovych

Discipline	System programming in the Unix environment
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, Object-oriented programming, Algorithms and calculation methods, System programming, Algorithms and data structures, Introduction to the Linux operating system, Architecture Computers, Software Engineering,
What will be studied?	Design and development of system programs for Unix-like systems. Programming takes place at the level of system calls (the corresponding functions in the standards of program development in the Unix environment), that is, at a low level of interaction with the kernel. Basic information on APIs in a Unix-like system (with an understanding of system call implementations in the kernel) for developing system programs. The discipline is not focused on system programming in any one specific implementation of a Unix-like operating system, that is, portable system programming will be studied. The course consists of the following topics: process environment, process management, advanced file handling, signal handling, interprocess communication (IPC), advanced I/O, terminal programming.

Why is it interesting/necessary to study?	This discipline should be studied by those who will develop system programs for Unix-like systems. Programming tasks in the discipline are performed in C or C++, but the acquired knowledge will be useful for developing system programs for Unix-like systems in other programming languages.
What will be learned?	Develop system programs for Unix-like systems in the C or C++ programming language for managing processes, working with files, signals, using advanced I/O, interprocess interactions (IPC), work with terminals.
How to use acquired knowledge and skills (competencies)?	The acquired knowledge can be used in the design and development of system programs for Unix-like systems, to support the source code of existing system programs for Unix-like systems, in the development of more effective applications programs
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory work
Semester control	Offset
Teacher	Sr. teacher Simonenko Andriy Valeriyovych

Discipline	Systems of automated design of computer systems
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian, English
Department	Computer Engineering
Prerequisites	Basic knowledge of: Higher mathematics, Programming, Discrete mathematics, Computer circuitry, Computer architecture
What will be studied?	<ul style="list-style-type: none"> – the basics of the structure of automated design systems (CAD) computer systems (CS); – the technology of designing specialized conveyor systems based on programmable logic integrated circuits (PLCs) using modern CAD; – design of specialized multiprocessor CPUs and their interfaces.
Why is it interesting/necessary to study?	<ul style="list-style-type: none"> – Modern PLD are almost the only elemental base that makes it possible to design the latest CSs, which differ in high performance, reliability, and energy consumption characteristics; – PLDs are used in medical devices, telecommunications, transport, data centers, the Internet of Things, military electronics, aerospace industry, to solve artificial

	<p>intelligence problems;</p> <ul style="list-style-type: none"> – there is a big shortage of professional personnel, both in Ukraine and in the world, who are able to design CS on PLD, whose salary is, as a rule, higher than the salary of an average programmer.
What will be learned?	<ul style="list-style-type: none"> – To compile parallel algorithms for high-performance calculations, – to program algorithms in VHDL language for implementation in hardware CSs, – to model and test CS projects, compile the CS description into PLD firmware.
How can you use acquired knowledge and skills?	<ul style="list-style-type: none"> – To analyze the suitability of PLD for performing computer tasks that require high performance; – to design high-performance CS on PLD according to the effective technology; – to perform modernization of existing high-performance CS on PLD; – to organize the execution of algorithms described in C, Python, on hardware accelerators on PLD. <p>Mastery of PLD programming technology is required in Ukraine in the fields of military equipment design, telecommunications, information protection, aircraft construction, including design of drones and abroad in these fields, in artificial intelligence systems, IoT, ADAS, robotics, design of custom microcircuits.</p>
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works
Semester control	Offset
Teacher	Doctor of Science, professor Sergienko Anatoly Mykhailovych

Discipline	Testing and quality control (QA) of embedded systems
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, Object-Oriented Programming, Algorithms and

	<p>calculation methods, System programming, Algorithms and Data structures, Computer architecture, Software engineering, Basics of parallel programming</p>
<p>What will be studied?</p>	<p>Basics of testing. Testing purposes at different stages of life product development cycle. Software development models. Basic testing levels and their relationships. Compilation of test documentation. Working with equipment based on BeagleBone Black platform. Independent creation of an embedded system, setting up an environment. Experience with embedded operating system testing in view of test environment setup, as part of which following questions are considered:</p> <ul style="list-style-type: none"> – Network troubleshooting; – Configuration of the Linux OS and computer network; – OS deployment (Linux kernel, U-Boot, BusyBox) for ARM Cortex A8 processor architecture by source code compilation; different approaches to flashing processor chips on the board; – Performing testing of own product based on BeagleBone Black platform; – Using Git and GitHub for version control; <p>Validation of functional and non-functional attributes software and hardware in self-created embedded system. Creating tests for verification and validation of embedded software and hardware in accordance with client requirements.</p>
<p>Why is it interesting/necessary to study?</p>	<p>In IT companies, great attention is paid to product quality, which is released to the market. The main reasons for the need for testing, which explain the special training of specialist testers:</p> <ul style="list-style-type: none"> – finding errors in software and hardware; – risk reduction for both users and the company; – reduction of development and maintenance costs; – productivity improvement. <p>Embedded software and hardware testing is an excellent approach to ensure safety in critical applications such as medical equipment, railways, aviation, transportation industry, etc. Large IT companies engaged in such developments invite specially trained testers to work. To train specialists for testing embedded systems, it is necessary to have basic knowledge in the sphere of design and operation of embedded systems, computer networks.</p>
<p>What will be learned?</p>	<ul style="list-style-type: none"> – To configure the Linux OS and computer network; – To deploy an OS (Linux kernel, U-Boot, BusyBox) for ARM Cortex A8 processor architecture by source code compilation; – Use different methods of flashing processor chips on the board. Use various interfaces, terminal emulation programs for communication and debugging of built-in devices; – Perform testing of own product based on BeagleBone Black platform; – Find errors in software and hardware embedded systems; – Troubleshoot computer network problems;

	<ul style="list-style-type: none"> – Compile test documentation; – Work in team; – Get basic training for QA specialist certification ISTQB.
How can you use acquired knowledge and skills?	Work in IT as a software and hardware tester. Continue training in the sphere of software and hardware development of embedded systems. Continue training in the sphere of software and hardware testing automation specialist. Acquired basic knowledge in the field of the basics of QA testing can be used in the field of software testing, as well as being the basis for independent preparation for the ISTQB specialist QA certification.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, practical and seminar classes, laboratory works
Semester control	Offset
Teacher	Doctor of Science, associate professor, professor Klymenko Iryna Anatolyivna

Discipline	Methodology and organization of scientific research
Level of higher education	First (Bachelor)
Course	4
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Probability theory, Algorithms and calculation methods, System programming, System software, Computer systems, Data structures and algorithms, Organization of databases, Programming, Computer architecture, Computer networks, Discrete mathematics
What will be studied?	<ul style="list-style-type: none"> – Methods of conducting scientific research; – Mathematical modeling methods; – Methods of displaying and transforming information.
Why is it interesting/necessary to study?	The discipline lays the foundations of independent scientific work starting with laboratory work, course and diploma design
What will be learned?	Create various models, methods of conducting scientific research
How can you use acquired knowledge and skills?	Plan scientific research, process the results of scientific research, process and analyze the obtained results

Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works
Semester control	Offset
Teacher	Ph.D., associate professor Selivanov Viktor Levovych

Discipline	Artificial intelligence methods and technologies
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Programming, System programming, Computer architecture, Mathematics, Probability theory
What will be studied?	The main theoretical principles of the organization, construction, structures and algorithms of intelligent embedded systems and real-time systems will be studied. Acquiring practical skills in developing schedulers, quantifying the real-time data processing capabilities of computer systems and specialized processors. Laboratory works are scheduled on the following topics: <ul style="list-style-type: none"> – Research and construction of correlation functions. – Applied problems of security and artificial intelligence. Neural network training. – Development of an intelligent scheduler of an embedded system.
Why is it interesting/necessary to study?	This course is designed for students who enjoy hands-on programming and problem-solving. Students will learn: <ul style="list-style-type: none"> – to program in the Python language for modeling planning disciplines; – we will additionally program in Java (Kotlin) and C. The course also covers interesting tasks from KPI-Open Olympiad programming. This will help to create effective algorithms in conditions of limited resources of embedded systems.
What will be learned?	Practical knowledge on the development of schedulers of embedded systems will be obtained. Knowledge and ability to evaluate the computing capabilities of embedded systems, based on relevant mathematical statements, and how to practically develop a scheduler using the most common scheduling methods.

How can you use acquired knowledge and skills?	After completing the course, students will be able to develop and research software and technical means for fast, including parallel, information processing in computer systems that have direct connections with real objects. Examples of such systems are control systems for moving objects, autopilots and car safety systems, avionics devices, and virtual reality systems.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works, independent work
Semester control	Offset
Teacher	Ph.D., associate professor

Discipline	Basics of data science
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	Basic knowledge of: Higher mathematics (mathematical analysis, linear algebra and analytical geometry), Probability theory and Mathematical statistics, Programming
What will be studied?	General concepts about the essence of data analysis, the application of the Bonferroni principle, the method of searching for nearest neighbors, document shingling. Principles of preserving the similarity of reduced sets, locally sensitive hashing and its application, distance measures. The concept of analysis of data flows of various nature and complexity and analysis of links. The market basket model and the A-Priori algorithm.
Why is it interesting/necessary to study?	Data science has significant practical implications for solving pressing everyday problems. The application of data science changes the way of life, work and thinking, since one of the conditions for the successful development of the world economy at the current stage is the ability to record and analyze accumulated data sets. Modern effective methods of working with data contribute to increasing the competitiveness of various spheres of economic activity, as well as to the specialists who possess these methods.
What will be learned?	Create models for data using generalization and feature extraction. Use the methods of calculating the Jakarta similarity of sets.

	<p>Apply distance measures to determine similarity, including cosine distance, edit distance, Hamming distance, location-sensitive functions.</p> <p>Analyze data streams by building a stream data model and filtering streams, count various elements in a stream, in particular, counting in a window and fading windows, use DGIM algorithms.</p> <p>Perform link analysis using PageRank, its effective calculation, calculate thematically sensitive PageRank, determine link spam.</p>
How can you use acquired knowledge and skills?	<ul style="list-style-type: none"> – Formation of initial data sets using the Bonferroni principle; – search for textually similar documents (plagiarism, mirror pages, articles from one source); – determining the similarity of online shopping and movie rating; – object recognition, comparison of fingerprints and newspaper articles; – creation of software systems for effective filtering of unwanted flow elements (incoming e-mail flows); – creation of teleport kits.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works, independent work
Semester control	Offset
Teacher	Doctor of Science, professor Novotarskyi Mykhailo Anatoliyovych

Discipline	Statistical methods of machine learning
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	<p>Basic knowledge of:</p> <p>Higher mathematics (mathematical analysis, linear algebra and analytical geometry), Probability theory and Mathematical statistics, Programming; skills in solving typical problems of these disciplines</p>
What will be studied?	<p>The main methods of machine learning for classification, clustering and regression problems (prediction), in particular, the principles of building some basic classifiers, problems that require to use statistical machine learning methods, algorithmic and computational approaches to their implementation with in-depth study of the Python programming language and using of TensorFlow libraries and THEANO.</p>

<p>Why is it interesting/necessary to study?</p>	<p>To obtain significant competitive advantages in the field of IT technologies, since machine learning is widely used in building search engines that learn to provide relevant results; models for generating targeted advertising within mass advertising campaigns; anti-spam software for filtering e-mail messages; software for detecting fraud on the Internet, in particular in the case of credit card transactions; pattern recognition and voice commands.</p> <p>Machine learning is also widely used in scientific applications such as bioinformatics, medicine, and astronomy.</p>
<p>What will be learned?</p>	<p>Knowledge of the principles of constructing feature vectors, decision rules and classification; main types of classifiers; principles of building linear classifiers; principles of building nonlinear classifiers; features of the selection of features of classification and pre-processing of data.</p> <p>The ability to choose the appropriate type of classifier depending on the problem to be solved; choose a set of features for classification and carry out preliminary data processing; apply algorithms for building a classifier based on a sample; perform calculations related to the training and operation of the classifier in the Python environment using the TensorFlow and THEANO libraries.</p>
<p>How can you use acquired knowledge and skills?</p>	<ul style="list-style-type: none"> – Selection, construction, training and use of basic classifiers when solving tasks; – independent work in modern software complexes; mastering a large amount of information; – setting tasks and conducting an experiment, programming tasks for data analysis; – visualization of the obtained results.
<p>Information support</p>	<p>Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.</p>
<p>Form of conduct classes</p>	<p>Lectures, laboratory works, independent work</p>
<p>Semester control</p>	<p>Offset</p>
<p>Teacher</p>	<p>Doctor of Science, professor Novotarskyi Mykhailo Anatoliyovych</p>

Discipline	Java programming language
Level of higher education	First (Bachelor)
Amount	4 ECTS credits
Language of teaching	Ukrainian
Department	Computer Engineering
Prerequisites	<p>Basic knowledge of:</p> <p>Higher mathematics (mathematical analysis, linear algebra and analytical geometry), Probability theory and Mathematical statistics, Programming; skills in solving typical problems of these disciplines</p>
What will be studied?	<p>The main methods of machine learning for classification, clustering and regression problems (prediction), in particular, the principles of building some basic classifiers, problems that require to use statistical machine learning methods, algorithmic and computational approaches to their implementation with in-depth study of the Python programming language and using of TensorFlow libraries and THEANO.</p>
Why is it interesting/necessary to study?	<p>To obtain significant competitive advantages in the field of IT technologies, since machine learning is widely used in building search engines that learn to provide relevant results; models for generating targeted advertising within mass advertising campaigns; anti-spam software for filtering e-mail messages; software for detecting fraud on the Internet, in particular in the case of credit card transactions; pattern recognition and voice commands.</p> <p>Machine learning is also widely used in scientific applications such as bioinformatics, medicine, and astronomy.</p>
What will be learned?	<p>Knowledge of the principles of constructing feature vectors, decision rules and classification; main types of classifiers; principles of building linear classifiers; principles of building nonlinear classifiers; features of the selection of features of classification and pre-processing of data.</p> <p>The ability to choose the appropriate type of classifier depending on the problem to be solved; choose a set of features for classification and carry out preliminary data processing; apply algorithms for building a classifier based on a sample; perform calculations related to the training and operation of the classifier in the Python environment using the TensorFlow and THEANO libraries.</p>
How can you use acquired knowledge and skills?	<ul style="list-style-type: none"> – Selection, construction, training and use of basic classifiers when solving tasks; – independent work in modern software complexes; mastering a large amount of information; – setting tasks and conducting an experiment, programming tasks for data analysis;

	– visualization of the obtained results.
Information support	Educational and working programs of the discipline, Point-Rating System, educational and methodical complex.
Form of conduct classes	Lectures, laboratory works, independent work
Semester control	Offset
Teacher	