



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ПРОГРАМНИХ СИСТЕМ та ПРИКЛАДНОЇ МАТЕМАТИКИ
КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І СПЕЦІАЛІЗОВАНИХ
КОМП'ЮТЕРНИХ СИСТЕМ



Ф-КАТАЛОГ

**вибіркових навчальних дисциплін
першого (бакалаврського) рівня вищої освіти
для освітньо-професійної програми
«Системне програмування та спеціалізовані комп'ютерні системи»
спеціальності F7 (123) Комп'ютерна інженерія**

Київ 2026

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ПРОГРАМНИХ СИСТЕМ та ПРИКЛАДНОЇ МАТЕМАТИКИ

ЗАТВЕРДЖЕНО

Методичною радою
КПІ ім. Ігоря Сікорського
(протокол № __ від «__» _____ 2026 р.)

Ф-КАТАЛОГ
ВИБІРКОВИХ НАВЧАЛЬНИХ ДИСЦИПЛІН
ЦИКЛУ ПРОФЕСІЙНОЇ ПІДГОТОВКИ
для здобувачів ступеня бакалавра
за освітньою програмою «Системне програмування та
спеціалізовані комп'ютерні системи»
за спеціальністю F7 Комп'ютерна інженерія
(вступ 2023, 2024 року)

УХВАЛЕНО:

Вченою радою факультету програмних систем
та прикладної математики
КПІ ім. Ігоря Сікорського
(протокол № 8 від « 26 » січня 2026 р.)

Київ – 2026

Розробники Ф-каталогу:

Тарасенко-Клятченко Оксана Володимирівна, канд. техн. наук, доцент, доцент кафедри системного програмування та спеціалізованих комп'ютерних систем ФПСМ.

Ф-каталог розглянуто та погоджено на засіданні кафедри системного програмування та спеціалізованих комп'ютерних систем ФПСМ, протокол № 8 від 06.01.2026 р.

Зміст

Інструкція користувачам каталогу	5	
Ф-Каталог – 2026 р.	6	
Анотації вибіркових дисциплін для 3 курсу	8	
5 семестр – потрібно обрати 12 кредитів	1. Бази даних та засоби управління	9
	2. Бази даних XML	10
	3. Системи управління базами даних XML	11
	4. Захист інформації в комп'ютерних системах	12
	5. Інтелектуалізовані методи захисту інформації	13
	6. Методи розпізнавання кібератак	14
	7. Периферійні пристрої	15
	8. Пристрої вводу-виводу	16
	9. Інтерфейси вводу-виводу	17
	10. Технології та інструменти розробки програмного забезпечення	18
6 семестр – потрібно обрати 20 кредитів	1. Моделювання та оптимізація комп'ютерних систем	19
	2. Modeling and optimization of computer systems	20
	3. Моделювання	21
	4. Архітектура комп'ютерів. Архітектура для програмістів	22
	5. Архітектура комп'ютерів. Мікроархітектура	23
	6. Computer architecture. Microarchitecture	24
	7. Основи проектування трансляторів	25
	8. Теорія формальних мов та синтаксичного аналізу	26
	9. Theory of formal language and parsing	27
	10. Операційні системи	28
	11. Введення до операційних систем	29
	12. Мережеві операційні системи	30
	13. Вебдизайн	31
	14. Технології XML у вебдизайні	32
	15. XML technologies in Web design	33

	<i>Анотації вибіркових дисциплін для 4 курсу</i>	34
	1. Комп'ютерні системи	35
	2. Багатопотокові обчислювальні системи	37
	3. Обчислювальні системи паралельної обробки даних	39
	4. Вступ до функціонального програмування	41
	5. Functional programming in Common Lisp	42
	6. Introduction to functional programming	43
	7. Архітектура комп'ютера. Апаратне забезпечення	44
	8. Архітектура комп'ютера. Архітектура вбудованих систем	45
	9. Computer architecture. Hardware	46
	10. Системне програмне забезпечення	47
	11. Розробка програмного забезпечення систем реального часу	49
	12. Програмне забезпечення розподілених операційних систем реального часу	51
	13. Комп'ютерний зір	53
	1. Комп'ютерне забезпечення телекомунікацій	54
	2. Комп'ютерна та телекомунікаційна техніка	55
	3. Телекомунікаційні технології	56
	4. Управління IT-проектами	57
	5. Комп'ютерні системи штучного інтелекта	58
	6. Штучні нейронні мережі	59
	7. Евристичні методи глобальної оптимізації	60
	8. Computer systems of artificial intelligence	61
<i>7 семестр – потрібно обрати 16 кредитів</i>		
<i>8 семестр – потрібно обрати 8 кредитів</i>		

Інструкція користувачам каталогу

1. Кількість і обсяг (у кредитах ЄКТС) навчальних дисциплін, які може обрати студент (вибіркових дисциплін) визначається навчальним планом відповідного року набору , а саме для III курсу – 32 кредити (5 семестр – 12 кредитів, 6 семестр – 20 кредитів), IV курсу – 24 кредити (7 семестр - 16 кредитів, 8 семестр – 8 кредитів). У навчальному плані зазначається також семестр, у якому викладається вибіркова дисципліна, форма семестрового контролю, види та обсяги навчальних занять.
2. Безпосередній вибір студентами дисциплін здійснюється за допомогою АІС «my.kpi.ua»: кожний студент особисто здійснює вибір дисциплін, які він бажає вивчати в наступному навчальному році, з урахуванням кількості вибіркових дисциплін, їх обсягу у кредитах ЄКТС та семестру вивчення, визначених навчальним планом (інформацію наведено на стор. 5).
3. У разі неможливості формування навчальних груп нормативної чисельності для вивчення певної вибіркової дисципліни, студентам надається можливість здійснити **повторний вибір**, приєднавшись до вже сформованих навчальних груп. Кафедра може надати можливість вивчати обрану дисципліну за допомогою індивідуальних консультацій з використанням змішаної форми навчання.
4. Студент не може двічі обрати одну й ту ж саму навчальну дисципліну.
5. Студент може обрати дисципліну відповідного обсягу для вивчення у відповідному семестрі з вибіркових каталогів інших ОП рівня бакалавр шляхом написання заяви завідувачу кафедри.
6. Якщо студент з 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 випускової кафедри вважатиме потрібними для оптимізації навчальних груп і потоків.
7. Обрані студентом навчальні дисципліни зазначаються у його індивідуальному навчальному плані, який формується системою «my.kpi.ua».
8. Більше інформації про порядок реалізації студентами права на вільний вибір навчальних дисциплін можна знайти у Положенні про порядок реалізації студентами права на вільний вибір навчальних дисциплін (<https://osvita.kpi.ua/node/185>).

Ф-Каталог – 2026 р.

Дисципліни для 3 курсу (вибір другокурсниками) (потрібно обрати 32 кредити)

Для вивчення на третьому курсі потрібно обрати вісім дисциплін (32 кредити):

- в п'ятому семестрі три дисципліни (кожна по 4 кредити),
- в шостому семестрі п'ять дисципліни (кожна по 4 кредити).

№	Назва навчальної дисципліни	Семестр	Кіл-ть кредитів	Семестрова атестація
1	Бази даних та засоби управління	5	4	Залік
2	Бази даних XML	5	4	Залік
3	Системи управління базами даних XML	5	4	Залік
4	Захист інформації в комп'ютерних системах	5	4	Залік
5	Інтелектуалізовані методи захисту інформації	5	4	Залік
6	Методи розпізнавання кібератак	5	4	Залік
7	Периферійні пристрої	5	4	Залік
8	Пристрої вводу-виводу	5	4	Залік
9	Інтерфейси вводу-виводу	5	4	Залік
10	Технології та інструменти розробки програмного забезпечення	5	4	Залік
10	Моделювання та оптимізація комп'ютерних систем	6	4	Залік
11	Modeling and optimization of computer systems	6	4	Залік
12	Моделювання	6	4	Залік
13	Архітектура комп'ютерів. Архітектура для програмістів	6	4	Залік
14	Архітектура комп'ютерів. Мікроархітектура	6	4	Залік
15	Computer architecture. Microarchitecture	6	4	Залік
16	Основи проєктування трансляторів	6	4	Залік
17	Теорія формальних мов та синтаксичного аналізу	6	4	Залік
18	Theory of formal language and parsing	6	4	Залік
19	Операційні системи	6	4	Залік
20	Введення до операційних систем	6	4	Залік
21	Мережеві операційні системи	6	4	Залік
22	Вебдизайн	6	4	Залік
23	Технології XML у вебдизайні	6	4	Залік
24	XML technologies in Web design	6	4	Залік

Дисципліни для 4 курсу (вибір третьокурсниками)
(потрібно обрати 24 кредити)

Для вивчення на четвертому курсі потрібно обрати шість дисциплін (24 кредити):

- в сьомому семестрі чотири дисципліни (кожна по 4 кредити);
- в восьмому семестрі дві дисципліни (кожна по 4 кредити).

№	Назва навчальної дисципліни	Семестр	Кіл-ть кредитів	Семестрова атестація
1	Комп'ютерні системи	7	4	Залік
2	Багатопотокові обчислювальні системи	7	4	Залік
3	Обчислювальні системи паралельної обробки даних	7	4	Залік
4	Вступ до функціонального програмування	7	4	Залік
5	Functional programming in Common Lisp	7	4	Залік
6	Introduction to functional programming	7	4	Залік
7	Архітектура комп'ютера. Апаратне забезпечення	7	4	Залік
8	Архітектура комп'ютера. Архітектура вбудованих систем	7	4	Залік
9	Computer architecture. Hardware	7	4	Залік
10	Системне програмне забезпечення	7	4	Залік
11	Розробка програмного забезпечення систем реального часу	7	4	Залік
12	Програмне забезпечення розподілених операційних систем реального часу	7	4	Залік
13	Комп'ютерний зір	7	4	Залік
14	Комп'ютерне забезпечення телекомунікацій	8	4	Залік
15	Комп'ютерна та телекомунікаційна техніка	8	4	Залік
16	Телекомунікаційні технології	8	4	Залік
17	Управління IT-проектами	8	4	Залік
18	Комп'ютерні системи штучного інтелекту	8	4	Залік
19	Штучні нейронні мережі	8	4	Залік
20	Евристичні методи глобальної оптимізації	8	4	Залік
21	Computer systems of artificial intelligence	8	4	Залік

Анотації вибірових дисциплін для 3 курсу

Дисципліна	Бази даних та засоби управління
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання програмування в межах початкових курсів профільного факультету
Що буде вивчатися	- способи організації сучасних баз даних; - функціонування систем управління базами даних; - створення клієнт-серверних додатків для обробки баз даних.
Чому це цікаво/треба вивчати	Будь-яка галузь бізнесу, науки, освіти, державного управління тощо вимагає вирішення різноманітних задач обробки та збереження інформації: систематизації, пошуку, сортування та інших, що розв'язуються завдяки базам даних. Тому сучасний фахівець в галузі ІТ технологій має володіти знаннями щодо організації подібних систем.
Чому можна навчитися (результати навчання)	- мати навички щодо ефективного подання інформації про предметну галузь в комп'ютерних системах; - вміти налаштовувати та програмувати системи управління базами даних; - створювати додатки, орієнтовані на обробку великих обсягів даних, зокрема у вебсистемах.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здобуті знання та уміння в галузі баз даних допоможуть студентам у розв'язанні широкого кола задач обробки інформації, необхідних для створення сучасного програмного забезпечення в рамках автоматизації процесів у різних предметних галузях.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Бази даних XML
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання програмування в межах початкових курсів профільного факультету
Що буде вивчатися	- інструментальні та мовні засоби XML; - функціонування систем управління базами даних XML; - створення клієнт-серверних додатків для обробки баз даних.
Чому це цікаво/треба вивчати	Інтерес до баз даних XML випливає із необхідності застосування XML у різних сферах ІТ-технологій: як засіб зберігання ієрархічних даних, серіалізації об'єктів, перетворення між різними типами документів, а також як структуру даних у програмуванні.
Чому можна навчитися (результати навчання)	- знати мови сімейства XML, їх синтаксичні аналізатори, засоби валідації, генерації та перетворення; - вміти взаємодіяти із базами даних XML (створювати записи даних, виконувати запити тощо); - створювати власні додатки, запрограмовані мовами XML для обробки даних та документів у форматі XML.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здобуті знання та уміння в галузі баз даних XML допоможуть студентам у розв'язанні задач, пов'язаних із обробкою великих обсягів слабо структурованих ієрархічних даних, будувати додатки типу вебсервісів XML та вміти автоматично формувати та аналізувати конфігураційні та журнальні файли різноманітних інформаційних систем.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Системи управління базами даних XML
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання програмування в межах початкових курсів профільного факультету
Що буде вивчатися	- системи управління базами даних XML та їх компоненти; - розробка додатків, орієнтованих на обробку XML-даних; - створення додатків з використанням мови XQuery.
Чому це цікаво/треба вивчати	Інтерес до систем управління базами даних XML впливає із необхідності обробки слабкоструктурованих даних у форматі XML, а також виконання таких операцій як індексація, тестування та обробка форм XML.
Чому можна навчитися (результати навчання)	- знати мови сімейства XML, їх синтаксичні аналізатори, засоби валідації, генерації та перетворення; - вміти створювати додатки мовою XQuery; - виконувати сервісні операції над базами даних XML.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Здобуті знання та вміння в галузі баз даних XML допоможуть студентам у розв'язанні задач, пов'язаних із обробкою слабкоструктурованої інформації у форматі XML, а також адміністрування XML-орієнтованих систем управління базами даних.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Захист інформації в комп'ютерних системах
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систе
Вимоги до початку вивчення	знання в межах повної загальної середньої освіти, основ програмування та вищої математики
Що буде вивчатися	- вимоги до засобів захисту інформації в комп'ютерних системах; - теоретичні положення створення засобів захисту інформації в області розподілу прав доступу, криптографії, стеганографії, захисту від мережових кібератак, комп'ютерних вірусів та спаму; інформації;
Чому це цікаво/треба вивчати	отримані знання будуть необхідними та корисними для кожного фахівця в області інформаційних технологій, як під час здійснення службових обов'язків, так і в повсякденному житті (наприклад, при використанні сучасних інформаційно-комунікаційних засобів – мережових месенджерів, електронної пошти, соціальних мереж, різного роду веб-сайтів та ін.)
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> • знати і розуміти нормативну базу захисту інформації в комп'ютерних системах; • знати і розуміти основні методи захисту інформації в комп'ютерних системах; • реалізовувати програмні механізми захисту інформації в комп'ютерних системах;
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>P4. Здатність забезпечувати захист інформації, що обробляється в комп'ютерних та кіберфізичних системах та мережах з метою реалізації встановленої політики інформаційної безпеки.</p> <p>P7. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.</p>
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Інтелектуалізовані методи захисту інформації
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання в межах повної загальної середньої освіти, основ програмування та вищої математики
Що буде вивчатися	- вимоги до засобів захисту інформації в комп'ютерних системах; - теоретичні положення створення інтелектуалізованих засобів захисту інформації; - технологію створення, налаштування і застосування нейронних мереж при розпізнаванні кібератак. веденні інформційних війн та в області біометричної аутентифікації.
Чому це цікаво/треба вивчати	отримані знання будуть необхідними та корисними для кожного фахівця в області інформаційних технологій, як під час здійснення службових обов'язків, пов'язаних з захистом інформації, так і службових обов'язків, пов'язаних з аналізом різноманітних бізнес процесів
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> • знати і розуміти нормативну базу захисту інформації; • знати технологію застосування нейронних мереж в області захисту інформації; • реалізовувати програмні механізми застосування нейронних мереж в області біометрії; • реалізовувати програмні механізми застосування нейронних мереж для розпізнавання кібератак;
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Р4. Здатність забезпечувати захист інформації, що обробляється в комп'ютерних та кіберфізичних системах та мережах з метою реалізації встановленої політики інформаційної безпеки.</p> <p>Р7. Здатність використовувати та впроваджувати нові технології, включаючи модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.</p>
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Методи розпізнавання кібератак
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання в межах повної загальної середньої освіти, основ програмування та вищої математики
Що буде вивчатися	- вимоги до засобів розпізнавання кібератак на комп'ютерні системи; - теоретичні положення створення інтелектуалізованих засобів розпізнавання кібератак; - технологія створення, налаштування засобів кібератак.
Чому це цікаво/треба вивчати	отримані знання будуть необхідними та корисними для кожного фахівця в області комп'ютерної інженерії під час здійснення службових обов'язків, пов'язаних із захистом інформації.
Чому можна навчитися (результати навчання)	- знати і розуміти нормативну базу в області розпізнавання кібератак інформації; - знати технологію застосування методів штучного інтелекту для розпізнавання кібератак; - реалізовувати програмні механізми застосування засобів штучного інтелекту для розпізнавання мережевих кібератак та розпізнавання комп'ютерних вірусів; розпізнавання мережевих кібератак на Інтернет-ресурси.
Як можна користуватися набутими знаннями і уміннями (компетентності)	P4. Здатність забезпечувати захист інформації, що обробляється в комп'ютерних та кіберфізичних системах та мережах з метою реалізації встановленої політики інформаційної безпеки. P7. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних та безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення.
Форма проведення занять	Лекції, лабораторні заняття.
Семестровий контроль	Залік

Дисципліна	Периферійні пристрої
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Периферійні пристрої» повинні передувати дисципліни «Комп'ютерна електроніка», «Програмування», «Комп'ютерна схемотехніка»
Що буде вивчатися	Базові набори інтегральних мікросхеми для побудови взаємодії комп'ютерних систем з периферійними пристроями; Функціональні особливості різноманітних периферійних пристроїв для взаємодії оператора з комп'ютерними системами; Алгоритми функціонування периферійних пристроїв та їх зв'язки з мікроконтролерами; Принципи побудови апаратної побудови спеціалізованих периферійних пристроїв для систем управління процесами; Методика програмування процесу обміну між периферійними пристроями та комп'ютерною системою; Методика розрахунку і побудови функціональних та електричних схем;
Чому це цікаво/треба вивчати	отриманні знання будуть необхідними та корисними для кожного програміста, оскільки спочатку викладається загальна структура організації взаємодії між різноманітними пристроями, що дає розуміння роботи з комп'ютерною системою; розглядаються особливості програмування взаємодії комп'ютерних систем з об'єктом на мові низького рівня.
Чому можна навчитися (результати навчання)	аналізувати електричні схеми периферійних пристроїв; розробляти програми керування периферійним пристроєм для комп'ютерних та мікропроцесорних систем; оцінювати складність побудови апаратно-програмного забезпечення систем управління;
Як можна користуватися набутими знаннями і вміннями (компетентності)	Вивчення дисципліни «Периферійні пристрої» надає досвід: розуміння взаємодії комп'ютерної системи з різноманітними пристроями для вводу та виводу інформації; проектувати спеціалізовані системи управління; розроблення апаратного та програмного забезпечення на базі мікроконтролерів; створення систем управління згідно технічного завдання; застосування різноманітних інтерфейсних з'єднань між елементами системи.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Пристрої вводу-виводу
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Пристрої вводу-виводу» повинні передувати дисципліни «Комп'ютерна електроніка», «Програмування», «Комп'ютерна схемотехніка»
Що буде вивчатися	Функціональні особливості пристроїв вводу; Функціональні особливості пристроїв виводу; Організація взаємодії пристроїв вводу-виводу через інтерфейсні з'єднання; Створення програмних продуктів для керування та обробки даних з пристроїв вводу-виводу; Методики розрахунку і побудови функціональних та електричних схем та особливості застосування мікроконтролерів, цифрових та аналогових сенсорів.
Чому це цікаво/треба вивчати	отриманні знання будуть необхідними та корисними для кожного програміста, оскільки спочатку викладається загальна структура організації між різноманітними пристроями, що дає розуміння взаємодії з комп'ютерною системою; розглядаються особливості програмування взаємодії комп'ютерних систем з об'єктом на мові низького рівня.
Чому можна навчитися (результати навчання)	аналізувати електричні схеми пристроїв вводу-виводу; оцінювати складність побудови апаратно-програмного забезпечення систем управління;
Як можна користуватися набутими знаннями і уміннями (компетентності)	Вивчення дисципліни «Пристрої вводу-виводу» надає досвід: Проектування вбудованих комп'ютерних систем; Розроблення апаратного та програмного забезпечення на базі мікроконтролерів; Створення систем управління згідно технічного завдання; Застосування різноманітних інтерфейсних з'єднань між елементами системи.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Інтерфейси вводу-виводу
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Інтерфейси вводу-виводу» повинні передувати дисципліни «Комп'ютерна електроніка», «Програмування», «Комп'ютерна схемотехніка»
Що буде вивчатися	Алгоритми функціонування інтерфейсів вводу-виводу по послідовному каналу; Алгоритми функціонування інтерфейсів вводу-виводу по паралельному каналу; Алгоритми функціонування спеціалізованих інтерфейсів вводу-виводу; Програмування процесу вводу-виводу для комп'ютерних систем та мікроконтролерів використовуючи різноманітні інтерфейси; Апаратні особливості організації інтерфейсів вводу-виводу; Методика розрахунку і побудови функціональних та електричних схем; Використання інтерфейсів вводу-виводу для побудови систем управління об'єктом.
Чому це цікаво/треба вивчати	отриманні знання будуть необхідними та корисними для кожного інженера системотехніка, оскільки вони дозволяють глибше розуміти процес взаємодії комп'ютерних систем об'єктом керування підключеного через відповідний інтерфейс, що полегшує процес написання програми керування; розглядаються особливості програмування взаємодії комп'ютерних систем з об'єктом на мові низького рівня.
Чому можна навчитися (результати навчання)	аналізувати електричні схеми та організовувати інтерфейси обміну з використанням базових контролерів вводу-виводу; оцінювати складність побудови апаратно-програмного забезпечення систем управління.
Як можна користуватися набутими знаннями і уміннями (компетентності)	досвід: проектування систем управління; розроблення апаратного та програмного забезпечення на базі мікроконтролерів; створення систем управління згідно технічного завдання; застосування різноманітних інтерфейсних з'єднань між елементами системи.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Технології та інструменти розробки програмного забезпечення
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС (54 год. аудиторної роботи, 66 год. СРС)
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання програмування в межах початкових курсів профільного факультету
Що буде вивчатися	програмні інструменти та технології, які використовуються при створенні сучасного програмного забезпечення
Чому це цікаво/треба вивчати	Кожен програміст, крім знання мов програмування, повинен вміти ще і користуватись допоміжними засобами та програмним забезпеченням (ПЗ), без яких неможливо уявити жоден сучасний проект зі створення програмних продуктів – таких, як системи контролю версій, системи збирання, системи безперервної інтеграції/безперервного розгортання, статичні аналізатори коду, різноманітні види тестування ПЗ, тощо. Всі ці інструменти і технології вивчаються на цьому предметі. Даний курс створений в колаборації зі спеціалістами компанії “Ajax Systems” і викладається її співробітниками
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - вміти користуватись системами контролю версій - знати, відрізнити і вміти застосовувати різні методології ведення проектів - навчитися застосовувати різні методи тестування ПЗ - вміти користуватись та налаштовувати системи безперервної інтеграції та розгортання - познайомитися з реальними прикладами проектів зі створення програмного забезпечення
Як можна користуватися набутими знаннями і уміннями (компетентності)	Знання і навички з цього курсу дають змогу впевнено користуватись основними сучасними засобами розробки ПЗ та ведення проектів.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Моделювання та оптимізація комп'ютерних систем
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліна базується на знаннях, отриманих студентами при вивченні таких дисциплін, як «Математичний аналіз», «Теорія електричних кіл та сигналів», «Програмування», «Комп'ютерна схемотехніка».
Що буде вивчатися	Курс дасть можливість навчитися аналізувати та проводити математичне моделювання складних процесів, що дає можливість оптимізувати затрати при побудові інформаційних систем.
Чому це цікаво/треба вивчати	Курс надасть знання, які дозволять аналізувати вимоги систем перетворення інформації, обирати методи моделювання цифрових схем, забезпечувати математичне моделювання складних систем перетворення інформації, виконувати програмування та моделювання динамічних процесів.
Чому можна навчитися (результати навчання)	Методам математичного моделювання складних систем перетворення інформації, способам моделювання динамічних процесів по експериментальним даним, алгоритмам моделювання динамічних процесів, принципам логічного моделювання, методикам імітаційного моделювання, особливостям різноманітних підходів до моделювання цифрових пристроїв.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Після вивчення курсу студент набуде: уміння: <ul style="list-style-type: none"> • аналізувати електричні схеми; • оцінювати складність моделювання цифрових пристроїв; досвід: <ul style="list-style-type: none"> • моделювання складних систем; • розроблення цифрових пристроїв; • моделювання динамічних процесів по експериментальним даним; • застосування різноманітних підходів до моделювання цифрових пристроїв.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, методичні вказівки по виконанню лабораторних робіт, слайди лекцій
Форма проведення занять	Лекції та лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Modeling and optimization of computer systems
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Англійська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліна базується на знаннях, отриманих студентами при вивченні таких дисциплін, як «Математичний аналіз», «Теорія електричних кіл та сигналів», «Програмування», «Комп'ютерна схемотехніка».
Що буде вивчатися	Курс дасть можливість навчитися аналізувати та проводити математичне моделювання складних процесів, що дає можливість оптимізувати затрати при побудові інформаційних систем.
Чому це цікаво/треба вивчати	Курс надасть знання, які дозволять аналізувати вимоги систем перетворення інформації, обирати методи моделювання цифрових схем, забезпечувати математичне моделювання складних систем перетворення інформації, виконувати програмування та моделювання динамічних процесів.
Чому можна навчитися (результати навчання)	Методам математичного моделювання складних систем перетворення інформації, способам моделювання динамічних процесів по експериментальним даним, алгоритмам моделювання динамічних процесів, принципам логічного моделювання, методикам імітаційного моделювання, особливостям різноманітних підходів до моделювання цифрових пристроїв.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Після вивчення курсу студент набуде: уміння: <ul style="list-style-type: none"> аналізувати електричні схеми; оцінювати складність моделювання цифрових пристроїв; досвід: <ul style="list-style-type: none"> моделювання складних систем; розроблення цифрових пристроїв; моделювання динамічних процесів по експериментальним даним; застосування різноманітних підходів до моделювання цифрових пристроїв.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, методичні вказівки по виконанню лабораторних робіт, слайди лекцій
Форма проведення занять	Лекції та лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Моделювання
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліна базується на знаннях, отриманих студентами при вивченні таких дисциплін, як <i>«Комп'ютерні системи штучного інтелекту»</i> , <i>«Паралельні та розподілені обчислення»</i> .
Що буде вивчатися	Курс дасть можливість навчитися аналізувати та проводити математичне та цифрове представлення і обробка фізичних процесів, що змінюються у часі.
Чому це цікаво/треба вивчати	Курс надасть знання, які дозволять для розв'язувати практичні задачі майбутньої професійної діяльності, пов'язаної з аналізом потреб розробки та експлуатацією сучасних інформаційно-обчислювальних систем та мереж.
Чому можна навчитися (результати навчання)	Основних методів математичного моделювання складних систем та фізичних процесів використовуючи сучасні програмні засоби (такі як Matcad, Matlab та інші) для ефективного дослідження та оцінювання таких систем і формування прогнозів ефективності їх впровадження; Методології послідовного аналізу варіантів для створення методів та алгоритмів вирішення задач дослідження та проектування складних систем, математичні моделі яких формалізуються в класах задач математичного програмування.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Після вивчення курсу студент матиме: Здатність до абстрактного мислення, аналізу та синтезу; Здатність вчитися і оволодівати сучасними знаннями; Здатність застосовувати знання у практичних ситуаціях; Здатність організації міжособистісної взаємодії; Здатність виявляти, ставити та вирішувати проблеми.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, методичні вказівки по виконанню лабораторних робіт, слайди лекцій
Форма проведення занять	Лекції та лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Архітектура комп'ютерів. Архітектура для програмістів
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Архітектура комп'ютерів. Архітектура для програмістів» повинні передувати дисципліни "Структури даних та алгоритми", "Програмування", "Системне програмування", "Комп'ютерна логіка", "Паралельне програмування"
Що буде вивчатися	<ul style="list-style-type: none"> - теоретичні відомості про архітектурні елементи сучасних комп'ютерів та зв'язки між ними; - основні поняття про модель комп'ютера, яку необхідно програмувати певною алгоритмічною мовою; - зв'язок між елементами архітектури та структурами даних та операціями мови програмування; - кластерні обчислення;
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> - знання архітектури комп'ютерів дасть змогу виконувати розробку складних і відповідальних програмних рішень, які мають особливі вимоги до швидкодії, надійності та захисту інформації; - це одні з основних знань, необхідних для того щоб бути професіоналом у галузі розробки програмного забезпечення;
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - одержання правильного уявлення про поняття архітектури комп'ютерів і зв'язки різних компонентів архітектури між собою; - одержання правильного уявлення про програмування як організації обчислювальних процесів в архітектурі;
Як можна користуватися набутими знаннями і уміннями (компетентності)	Студент отримає знання і досвід розробки програмних рішень з урахуванням особливостей архітектури комп'ютерів, що дасть змогу створювати оптимізовані для певної архітектури рішення, що характеризуватимуться високими показниками швидкодії, надійності, захисту інформації та енергоефективності.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Архітектура комп'ютерів. Мікроархітектура
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Архітектура комп'ютерів. Мікроархітектура» повинні передувати дисципліни "Структури даних та алгоритми", "Програмування", "Системне програмування", "Периферійні пристрої", "Схемотехніка (частина 1)", "Комп'ютерна логіка", "Паралельне програмування",
Що буде вивчатися	<ul style="list-style-type: none"> - теоретичні відомості про архітектурні елементи сучасних комп'ютерів та зв'язки між ними; - способи виконання архітектури рівня системи команди в апаратурі; - способи реалізації архітектури в ПЛІС; - способи реалізації архітектури в різних відомих архітектурах мікропроцесорів (Intel x86, ARM, RISC-V, та ін.)
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> - знання архітектури комп'ютерів дасть змогу виконувати розробку складних і відповідальних програмно-апаратних рішень, які мають особливі вимоги до швидкодії, надійності, захисту інформації та енергоефективності; - це одні з основних знань, необхідні для того щоб бути професіоналом у галузі розробки програмного та апаратного забезпечення;
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - одержання правильного уявлення про поняття архітектури комп'ютерів і зв'язки різних компонентів архітектури між собою; - одержання правильного уявлення про розробку мікроархітектур комп'ютерів;
Як можна користуватися набутими знаннями і уміннями (компетентності)	Студент отримає знання і досвід розробки спеціалізованих архітектур, що дасть змогу створювати оптимізовані для певної задачі програмно-апаратні засоби, що характеризуються високими показниками швидкодії, надійності, захисту інформації та енергоефективності.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Discipline	Computer architecture. Microarchitecture
Level of higher education	First (bachelor)
Course	3
Teaching hours	4 credits ECTS 54 год. аудиторної роботи, 66 год. CPC
Language of study	English
Department	System programming and specialized computer systems
Prerequisites	The discipline "Computer architecture. Microarchitecture" must be preceded by the disciplines "Data structures and algorithms", "Programming", "System programming", "Peripheral devices", "Schematic engineering (first semester)", "Computer logic", "Parallel programming"
What will be studied	<ul style="list-style-type: none"> - theoretical information about the architectural elements of modern computers and the connections between them; - ways of implementing the architecture of the command system level in the equipment; - methods of implementing architecture in FPGA; - ways of implementing the architecture in various known microprocessor architectures (Intel x86, ARM, RISC-V, etc.)
Why is it of interest / necessary to study	<ul style="list-style-type: none"> - knowledge of computer architecture will enable the development of complex and responsible software and hardware solutions that have special requirements for speed, reliability, information protection, and energy efficiency; - this is one of the basic knowledge needed to be a professional in the field of software and hardware development;
What you can learn (studying outcomes)	<ul style="list-style-type: none"> - obtaining a correct understanding of the concept of computer architecture and the connections between various components of the architecture; - getting the right idea about the development of computer microarchitectures;
How to use the acquired knowledge and skills (competencies)	The student will gain knowledge and experience in the development of specialized architectures, which will allow the creation of software and hardware solutions optimized for a specific task, characterized by high performance, reliability, information protection, and energy efficiency.
Information support	Training and working programs of the discipline, RSA, a set of educational and methodological materials
Form of classes	Lectures, lab classes
Semester control	Test

Дисципліна	Основи проєктування трансляторів
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Основи проєктування трансляторів» повинні передувати дисципліни "Структури даних та алгоритми", "Програмування" та "Системне програмування"
Що буде вивчатися	Всі основні стадії розробки транслятора та необхідна для цього теорія: <ul style="list-style-type: none"> - основи теорії формальних мов; - лексичний аналіз мов програмування; - синтаксичний аналіз мов програмування; - генерація коду для мов програмування.
Чому це цікаво/треба вивчати	- отриманні знання будуть необхідними та корисними для кожного системного програміста, оскільки викладається як теорія, так і всі стадії розробки одного з основних інструментів програміста - транслятора.
Чому можна навчитися (результати навчання)	- читати та писати граматики мов програмування; - розроблювати лексичні аналізатори; - розроблювати синтаксичні аналізатори; - розроблювати генератори коду.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Студент отримає: <ul style="list-style-type: none"> - розуміння процесу трансляції програм; - досвід розробки реально працюючого транслятора, створивши всі основні його частини: лексичний аналізатор, синтаксичний аналізатор, генератор коду. <p>Лексичні та синтаксичні аналізатори, крім того, що вони є складовими частинами компіляторів та інтерпретаторів, вони також є невід'ємними частинами засобів статичного аналізу коду, які широко використовуються для рефакторингу коду та покращення вже існуючих програмних систем.</p> <p>Ці знання та уміння забезпечують цінність та конкурентноздатність спеціаліста на ринку системного програмування.</p>
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Теорія формальних мов та синтаксичного аналізу
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Теорія формальних мов та синтаксичного аналізу» повинні передувати дисципліни "Математичний аналіз", "Прикладна теорія цифрових автоматів", "Структури даних та алгоритми", "Програмування" та "Системне програмування".
Що буде вивчатися	<ul style="list-style-type: none"> - розділи теорії формальних мов; - розробка граматики мов програмування; - різні методи та алгоритми лексичного аналізу мов програмування; - різні методи та алгоритми теорії синтаксичного аналізу, їх переваги та недоліки, відповідність різним класам формальних мов.
Чому це цікаво/треба вивчати	Оскільки ця дисципліна є більш науково зорієнтованою, то: <ul style="list-style-type: none"> - отримані знання будуть необхідними та корисними в першу чергу для студентів, які мають схильність до теоретичної та наукової роботи; - студенти навчатися глибоко розбиратися в процесі аналізу мов програмування.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - розбиратися в типах та класах граматики формальних мов; - розуміти процес лексичного аналізу; - вивчити різні методи та алгоритми синтаксичного аналізу мов програмування, їх переваги та недоліки, відповідність різним класам формальних мов.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Студент отримає досвід і зможе правильно використовувати знання теорії формальних мов та синтаксичного аналізу на практиці розробки лексичних та синтаксичних аналізаторів різних типів та класів.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Theory of Formal Languages and Parsing
Рівень ВО	First (bachelor)
Курс	3
Обсяг	4 credits of ECTS 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	English
Кафедра	System programming and specialized computer systems
Вимоги до початку вивчення	«Theory of formal languages and parsing» subject should be preceded by "Mathematic analysis", "Applied theory of finite-state machines ", "Data structures and Algorithms", "Programming" and "System programming" subjects.
Що буде вивчатися	<ul style="list-style-type: none"> - Chapters of formal languages theory; - Programming languages grammars creation; - Various methods and algorithms of lexical analysis of programming languages; - Various methods and algorithms of parsing theory, pros. and cons. of these methods, correspondence of the methods to formal languages classes.
Чому це цікаво/треба вивчати	<p>Because this subject is more theoretical then:</p> <ul style="list-style-type: none"> - acquired knowledge will be necessary and useful mostly for students who have predisposition to theoretical and scientific work; - students will be taught to understand parsing processes deeply.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - understand and use various types and classes of formal grammars; - understand lexical analysis processes; - study various parsing methods and algorithms of programming languages, their pros. and cons., correspondence of the methods to formal languages classes.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Student will acquire experience and will be able to use his knowledge of formal languages and parsing theory for solving practical tasks of scanners and parsers for grammars of different types and classes.
Інформаційне забезпечення	Teaching and Work programs of the subjects, Rating System of Estimation, Set of teaching-methodical documents.
Форма проведення занять	Lectures, Laboratory lessons
Семестровий контроль	Test

Дисципліна	Операційні системи
Рівень ВО	Бакалаври
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліна базується на знаннях, отриманих студентами при вивченні наступних дисциплін: «Об'єктно-орієнтоване програмування», «Паралельне програмування», «Системне програмування».
Що буде вивчатися	В результаті вивчення даного курсу студенти мають опанувати: призначення та функції операційних систем; еволюцію операційних систем у зв'язку з розвитком обчислювальної техніки; архітектуру операційних систем; відомості про мультипрограмування; методи планування процесів та потоків; засоби мультипрограмування на принципах переривань; основні засоби синхронізації процесів та потоків; проблему розподілу ресурсів та запобігання тупиків; алгоритми планування та планувальники; організацію та засоби управління пам'яттю; принципи побудови вводу-виводу інформації; файлові системи: логічна та фізична організації; порівняльний аналіз найбільш поширених методів організації сучасних файлових систем; практичні навички з реалізації типових механізмів побудови ОС при вирішенні конкретних задач.
Чому це цікаво/треба вивчати	Вивчення матеріалів курсу дозволить у подальшому: виконувати порівняльний аналіз найбільш поширених методів організації сучасних файлових систем; застосовувати практичні навички з реалізації типових механізмів побудови ОС при вирішенні конкретних задач; використати методи проектування складних програмних систем, до яких належать операційні системи.
Чому можна навчитися (результати навчання)	По результатах навчання можна буде виконувати : проектування основних елементів-програмних складових операційних систем; розробку алгоритмів планування та синхронізацію процесів, забезпечення їх реалізації на базі мікроконтролерів; створювати операційні системи управління згідно технічного завдання.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здобуті знання та уміння в галузі операційних систем та баз даних допоможуть майбутнім спеціалістам у розв'язанні широкого кола задач щодо створення як спеціалізованих операційних систем так і систем загального призначення.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції та лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Введення до операційних систем
Рівень ВО	Бакалаври
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліна базується на знаннях, отриманих студентами при вивченні наступних дисциплін: «Програмування», «Структури даних і алгоритми», «Об'єктно-орієнтоване програмування», «Паралельне програмування».
Що буде вивчатися	Загальні принципи побудови та функціонування операційних систем; Архітектура операційних систем; Способи побудови окремих функціональних компонентів операційних систем та їх реалізація в сучасних ОС.
Чому це цікаво/треба вивчати	Знання загальних принципів побудови операційних систем, як складних програмних систем, способів побудови функціональних компонентів ОС сприяють як кращому розумінню місця та ролі ОС в загальній ієрархії програмного забезпечення та можливостей їх використання при вирішенні прикладних задач, так і використанню для аналізу, проектування, розробки та імплементації програмного забезпечення системного і прикладного призначення.
Чому можна навчитися (результати навчання)	Знати: - архітектуру операційних систем; - функції та способи побудови складових операційних систем; - методи та алгоритми планування процесів та потоків; - основні засоби синхронізації процесів та потоків; - способи організації та засоби управління пам'яттю; - принципи та способи побудови системи вводу-виводу; - логічну та фізичну організацію файлових систем; - загальні принципи мережної підтримки в ОС; - основні завдання забезпечення безпеки в ОС. Мати практичні навички з проектування та реалізації типових механізмів та програмних елементів ОС.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здобуті знання та уміння в галузі операційних систем слугують подальшому успішному опануванню фахових дисциплін і допоможуть майбутнім спеціалістам у проведенні аналізу, проектуванні, розробці та імплементації програмного забезпечення системного і прикладного призначення при вирішенні конкретних інженерних задач різного ступеня складності з використанням методології системного програмування, зокрема для систем на базі мікроконтролерів і спеціалізованих КС.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення: Операційні системи. Навчальний посібник. http://ela.kpi.ua/handle/123456789/29600 Операційні системи. Лабораторний практикум. http://ela.kpi.ua/handle/123456789/25434
Форма проведення занять	Лекції та лабораторні заняття -
Семестровий контроль	Залік

Дисципліна	Мережеві операційні системи
Рівень ВО	Бакалаври
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліна базується на знаннях, отриманих студентами при вивченні наступних дисциплін: «Об'єктно-орієнтоване програмування», «Паралельне програмування», «Системне програмування».
Що буде вивчатися	ОС та задачі адміністрування. ОС Windows, Server, Linux. ПЗ автоматизованих інформаційних систем. Аналіз продуктивності ОС Сценарії BASH та PowerShell. Керування користувачами. Аналіз завантаженості системних ресурсів та мережі. Забезпечення безпеки в ОС Продуктивність та надійність веб-серверів. Продуктивність та надійність серверів баз даних.
Чому це цікаво/треба вивчати	Вивчення матеріалів курсу дозволить у подальшому виконувати адміністрування мереж, застосовувати практичні навички з реалізації типових механізмів побудови ОС при вирішенні конкретних задач; використати методи проектування складних програмних систем, до яких належать мережеві операційні системи, керувати обліковими записами
Чому можна навчитися (результати навчання)	По результатах навчання можна зрозуміти системи, засвоїти сутність задач адміністрування; ознайомитись з основними ОС, їх особливостями та характеристиками; засвоїти поняття ПЗ автоматизованих систем; відслідковувати процес завантаження ОС; засвоїти процес аналізу продуктивності ОС; ознайомитись з призначенням, основними функціями, командами та утилітами сценарію BASH; засвоїти призначення, основні функції, команди та утиліти сценарію PowerShell; навчитись управляти обліковими записами і ресурсами в середовищі Microsoft Windows Server; засвоїти основні характеристики жорстких дисків, файлових систем сучасних ОС, фактори впливу на завантаженість мережі; вивчити критерії визначення безпеки комп'ютерних систем та методи їх забезпечення.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Дисципліна передбачає формування практичних навичок управління параметрами завантаження операційної системи, виконання конфігурування апаратних пристроїв, автоматизації системних операцій, проектування і реалізації системних програмних засобів, формування знань про основні завдання адміністрування і способи їх виконання в досліджуваних операційних системах.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції та лабораторні зайняття
Семестровий контроль	Залік

Дисципліна	Вебдизайн
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Вебдизайн» повинні передувати дисципліни "Структури даних та алгоритми" та "Програмування"
Що буде вивчатися	<ul style="list-style-type: none"> - технології створення веб-інтерфейсу користувача - способи створення адаптивної розмітки вебсторінок; - засоби створення інтерактивних вебдодатків за допомогою мов Javascript та Typescript з використанням сучасних бібліотек та фреймворків, зокрема VueJS.
Чому це цікаво/треба вивчати	Веб-орієнтований інтерфейс користувача є найбільш популярним способом організації взаємодії людини та сучасного програмного забезпечення. З іншого боку, знання одних з найпопулярніших мов програмування Javascript та Typescript відкриває унікальні можливості щодо оволодіння конкурентними компетенціями на ринку працевлаштування.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - створювати візуальний дизайн сучасних веб-орієнтованих додатків - розроблювати інтерактивні та адаптивні вебдодатки - вміти користуватись середовищами розробки, бібліотеками та супутніми мовами програмування, а також тестувати та інтегрувати фрагменти інтерфейсів користувача у повноцінні додатки.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здобуті знання та уміння допоможуть студентам у розв'язанні широкого кола задач щодо створення веб-орієнтованих додатків різноманітного призначення: соціальних мереж, пошукових систем, офісних додатків, утиліт-віджетів тощо.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Технології XML у вебдизайні
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Технології XML у вебдизайні» повинні передувати дисципліни "Структури даних та алгоритми" та "Програмування"
Що буде вивчатися	<ul style="list-style-type: none"> - сімейство мов XML (XPath, XSLT та XQuery) для створення вебдодатків - документна модель вебдокумента (DOM), мова розмітки XHTML та мова стильового оформлення сторінок CSS - XML-орієнтовані засоби мови Javascript щодо маніпулювання XHTML DOM
Чому це цікаво/треба вивчати	XML-технології є невід'ємною складовою технологій розробки вебдодатків, зокрема, вони широко застосовуються у програмних бібліотеках, орієнтованих на взаємодію з веббраузером. Серед найпоширеніших задач використання XML у Web є створення, модифікація, а також перетворення вебдокументів для виведення на різних пристроях таких, як екран монітора, планшет або смартфон.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - створювати шаблони сторінок сучасних веб-орієнтованих додатків - розроблювати інтерактивні та універсальні вебдодатки - вміти користуватись середовищами розробки, бібліотеками та супутніми мовами програмування щодо застосування XML у веброботці.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здобуті знання та уміння допоможуть студентам у розв'язанні широкого кола задач щодо створення веб-орієнтованих додатків різноманітного призначення: соціальних мереж, пошукових систем, офісних додатків, утиліт-віджетів тощо.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Discipline	XML technologies in Web design
The level of VO	First (bachelor's)
Course	3
Amount	4 ECTS credits 54 год.аудиторної роботи, 66 год. СРС
Language of teaching	Ukrainian
Department	System programming and specialized computer systems
Requirements for the beginning of the study	The discipline "XML technologies in Web-design" should follow disciplines "Data structures and algorithms" and "Programming"
What will be studied	<ul style="list-style-type: none"> - XML family of languages (XPath, XSLT and XQuery) for creating Web applications - Web-oriented document object model (DOM), XHTML markup language, and CSS page styling language - XML-oriented Javascript tools for manipulating the XHTML DOM
Why it is interesting / necessary to study	XML-technologies are an integral part of Web application development technologies, in particular, they are widely used in software libraries focused on interaction with a Web-browser. Among the most common tasks of using XML in the Web are the creation, modification, and transformation of Web documents for output on various devices such as a monitor screen, tablet or smartphone.
Why you can learn (learning outcomes)	<ul style="list-style-type: none"> - create page templates for modern Web-oriented applications - develop interactive and universal Web-applications - be able to use development environments, libraries and related programming languages for the use of XML in Web development.
How to use the acquired knowledge and skills (competencies)	The acquired knowledge and skills will help students in solving a wide range of problems for creating Web-oriented applications for various purposes such as social networks, search engines, office applications, widget utilities, etc.
Information support	Training and working programs of the discipline, RSO, a set of educational and methodological support
Form of classes	Lectures, laboratory classes
Semester control	Test

Анотації вибірових дисциплін для 4 курсу

Дисципліна	Комп'ютерні системи
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Знати прикладну теорію цифрових автоматів, комп'ютерну арифметику, комп'ютерну логіку, дискретну математику, операційне числення, теорію ймовірностей та математичну статистику в обсязі, необхідному для розробки та використання сучасних обчислювальних систем
Що буде вивчатися	Дисципліна орієнтована на вивчення архітектур багатопроцесорних та багатомашинних комп'ютерних систем, сфери їх застосування, особливостей структурної організації. Також розглядаються питання, пов'язані із забезпеченням високої продуктивності сучасних обчислювальних систем паралельної організації та високопродуктивні обчислення на їх базі. Особливу увагу приділено вивченню організації кластерних систем та елементів кластеризації у рамках побудови сучасних швидкодіючих обчислювальних систем.
Чому це цікаво/треба вивчати	Сучасні комп'ютерні системи є необхідним механізмом, завдяки якому можна вирішувати різноманітні питання, а саме опрацьовувати транзакції у режимі реального часу, створювати сховища даних для організації систем підтримки прийняття рішень, будувати системи для глобальних корпоративних обчислень, здійснювати Big Tasks (прогнозування погоди, клімату, глобальних змін у атмосфері; науки про матеріали; побудова напівпровідникових пристроїв; розробка фармацевтичних препаратів; астрономія; логістика; розпізнавання та синтез мови; розпізнавання зображень, вивчення глобального потепління клімату, вплив новітніх вірусів і т. ін.). Тому вкрай необхідно знати, за якими принципами створюються сучасні високопродуктивні комп'ютерні системи, яким чином забезпечується їх надійність та живучість, які сучасні тести та рейтинги ефективності високопродуктивних комп'ютерних систем існують, які є перспективи розвитку цієї галузі. Як можна спроектувати комп'ютерну систему із заданими показниками ефективності. Як здійснити це максимально економічним шляхом, виходячи із специфіки математичних задач, що потребують вирішення. Однорідні та неоднорідні обчислювальні системи, кластерні реалізації, мережеві рішення, деякі питання організації сучасних квантових обчислювальних систем також розглядаються під час вивчення цього цікавого курсу. Особливої уваги приділено питанням і шляхам підвищення продуктивності обчислювальних систем з використанням конвеєрної обробки інформації та розшарування пам'яті у рамках побудови класичних та сучасних комп'ютерних систем.
Чому можна навчитися	– на основі знання основних тенденцій розвитку паралельних обчислювальних систем вирішувати питання створення нових

(результати навчання)	нетрадиційних варіантів побудови високопродуктивних і надпродуктивних засобів обчислювальної техніки; – базуючись на вивчення основних методів і засобів організації вводу-виводу даних, вирішувати питання побудови ефективних підсистем вводу-виводу даних з обліком продуктивності процесорних елементів системи
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> - розуміти шляхи і способи побудови як комп'ютерних систем паралельної архітектури, так і в подальшому комп'ютерних мереж, зрозуміти саму ідеологію розпаралелювання на рівні архітектури; - зв'язати у єдиний ансамбль методи і принципи паралельного програмування та методи і принципи побудови систем паралельної архітектури; - вільно орієнтуватись у світі високопродуктивних комп'ютерних систем та високопродуктивних обчислень на базі таких систем. Знати світові рейтинги ефективності, слідкувати за подіями у світі паралельних обчислень й паралельних суперкомп'ютерів; - набути здатність проектувати, розробляти та використовувати засоби реалізації комп'ютерних систем; - мати здатність оцінювати та враховувати економічні, соціальні, технологічні та екологічні фактори на всіх етапах життєвого циклу комп'ютерних систем; - мати здатність до аналізу, синтезу і оптимізації комп'ютерних систем з використанням математичних моделей і методів.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, підручник (електронне видання). Відзнятий матеріал та закінчується монтаж онлайн курсу «Загальна теорія розвитку» для платформи «Прометеус» (https://courses.prometheus.org.ua/courses/KPI/)
Семестровий контроль	Залік

Дисципліна	Багатопотокові обчислювальні системи
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Вивченню дисципліни передують такі курси як дискретна математика, прикладна теорія цифрових автоматів, лінійна алгебра та аналітична геометрія, диференціальне та інтегральне числення, теорія функцій багатьох змінних, теорія рядів, диференціальні рівняння для функції однієї та багатьох змінних, теорія ймовірностей та математична статистика.
Що буде вивчатися	Архітектури багатопотокових обчислювальних систем, сфери їх застосування, особливостей структурної організації. Також розглядаються питання, пов'язані із забезпеченням високої продуктивності сучасних багатопотокових обчислювальних систем та катастрофостійкі рішення. Надаються алгоритми побудови багатопотокових систем, які можуть із часом доволно нарощуватись. Описуються основні принципи побудови, архітектурна, функціональна та структурна організація багатопотокових обчислювальних систем, основні пристрої та ланки багатопотокових обчислювальних систем та деяких телекомунікаційних мереж. Розглядається їх функціонування, ефективність та перспективи розвитку.
Чому це цікаво/треба вивчати	Курс спрямований на вивчення архітектур багатопотокових комп'ютерних систем, сфери їх застосування, особливостей структурної організації. Також розглядаються питання, пов'язані із забезпеченням високої продуктивності сучасних багатопотокових обчислювальних систем та високопродуктивні обчислення на їх базі. Шляхи підвищення продуктивності за рахунок збільшення кількості апаратно підтриманих потоків управління порядку десятків тисяч для здійснення паралельних додатків. Розробники таких багатопотокових систем, орієнтованих на забезпечення високої пропускнуєї спроможності, намагаються не максимально підвищити однопотокову продуктивність, а до забезпечення можливості розпаралелювання із використанням великої кількості потоків управління, що апаратно підтримуються.
Чому можна навчитися (результати навчання)	- проводити системний аналіз багатопотокових систем та обґрунтовувати вибір структури, алгоритмів та способів передачі інформації в таких системах та технології їх побудови; - аналізувати, синтезувати і оптимізувати багатопотокові системи та використовувати математичні моделі та методи для їх побудови та розгортання
Як можна користуватися набутими знаннями і уміннями (компетентності)	- оцінювати та враховувати економічні, соціальні, технологічні та екологічні фактори на всіх етапах життєвого циклу багатопотокових систем; - аналізувати, синтезувати і оптимізувати багатопотокові системи з використанням математичних моделей і методів;

	- проектувати, розробляти та використовувати засоби реалізації багатопотокових обчислювальних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, підручник (електронне видання). Відзнятий матеріал та закінчується монтаж онлайн курсу «Загальна теорія розвитку» для платформи «Прометеус» (https://courses.prometheus.org.ua/courses/KPI/)
Семестровий контроль	Залік

Дисципліна	Обчислювальні системи паралельної обробки даних
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Знання теорії рядів, диференціальних рівнянь для функції однієї та багатьох змінних, теорії ймовірностей та математичної статистики, мережевих інформаційних технологій, прикладної теорії цифрових автоматів, функціонального програмування, паралельного програмування в обсязі, необхідному для розробки та використання інформаційних систем, технологій та комунікація задля побудови сучасних обчислювальних систем паралельної архітектури
Що буде вивчатися	Дисципліна орієнтована на вивчення сучасних та і класичних комп'ютерних систем з паралельною архітектурою, сфери їх застосування, особливостей структурної організації, питання забезпечення високої продуктивності подібних систем. Питання паралелізму, який є притаманним практично всім типам сучасних обчислювальних систем та його важлива місія у справі організації високопродуктивних обчислень.
Чому це цікаво/треба вивчати	Курс спрямований на вивчення архітектур паралельних комп'ютерних систем, зокрема систем класу SMP, MPP, NUMA. Питання і принципи побудови сучасних кластерних систем також розглядаються досить ретельно. Підвищення швидкодії систем і її вплив на спроможність реалізації задач Big Data за допомогою систем з паралельною архітектурою. Сучасні комп'ютерні системи є необхідним механізмом, завдяки якому можна вирішувати різноманітні питання, а саме опрацьовувати транзакції у режимі реального часу (OLTP), створювати сховища даних для організації систем підтримки прийняття рішень, будувати системи для глобальних корпоративних обчислень, здійснювати Grand Challenges (прогнозування погоди, клімату, глобальних змін у атмосфері; науки про матеріали; побудова напівпровідникових пристроїв; розробка фармацевтичних препаратів; астрономія; логістика; розпізнавання та синтез мови; розпізнавання зображень і т. ін.). Як можна спроектувати паралельну комп'ютерну системи із заданими показниками ефективності. Як здійснити це максимально економічним шляхом, виходячи із специфіки математичних задач, що потребують вирішення. Яким чином здійснюється побудова математичної моделі задля подальшого її опрацювання й оптимізації за допомогою паралельних обчислювальних систем.
Чому можна навчитися (результати навчання)	- проводити системний аналіз паралельних обчислювальних систем та обґрунтовувати вибір структури, алгоритмів та способів передачі інформації в таких системах та технології їх побудови; - аналізувати, синтезувати і оптимізувати паралельні обчислювальні системи та використовувати математичні моделі та методи для їх побудови та розгортання

<p>Як можна користуватися набутими знаннями і уміннями (компетентності)</p>	<ul style="list-style-type: none"> - проектувати, розробляти та використовувати засоби реалізації паралельних систем; - оцінювати та враховувати економічні, соціальні, технологічні та екологічні фактори на всіх етапах життєвого циклу паралельних систем; - аналізувати, синтезувати й оптимізувати паралельні системи з використанням математичних моделей і методів; - мати здатність проектувати, розробляти та використовувати засоби реалізації паралельних обчислювальних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).
<p>Інформаційне забезпечення</p>	<p>Навчальна та робоча програми дисципліни, PCO, підручник (електронне видання). Відзнятий матеріал та закінчується монтаж онлайн курсу «Загальна теорія розвитку» для платформи «Прометеус» (https://courses.prometheus.org.ua/courses/KPI/)</p>
<p>Семестровий контроль</p>	<p>Залік</p>

Дисципліна	Вступ до функціонального програмування
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 54 год аудиторних занять, 66 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Вступ до функціонального програмування» повинні передувати дисципліни "Структури даних та алгоритми", "Програмування"
Що буде вивчатися	<ul style="list-style-type: none"> - основні принципи функціонального програмування, його переваги та недоліки на прикладі мови програмування Common Lisp; - види рекурсії, та її застосування при вирішенні простих завдань конструктивним способом; - підходи до роботи з рекурсивними структурами даних, такими як однозв'язні списки; - поняття функцій вищого порядку та їхнє застосування для вирішення різних задач.
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> - парадигма функціонального програмування більшою чи меншою мірою реалізується в більшості сучасних мовах програмування, що створює потребу у її вивченні; - опанування функціонального програмування дає змогу розширити інструментарій розробника, покращити якість програмного коду, який він пише; - функціональний підхід до програмування дає змогу формалізувати задачі і розглядати їх в рамках певного математичного апарату, що значно спрощує аналіз таких задач.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - застосовувати рекурсію та працювати з рекурсивними структурами даних; - реалізовувати програми із застосуванням конструктивного підходу; - працювати з функціями як з об'єктами, в тому числі застосовувати типові функції вищого порядку, такі як map, filter, reduct, тощо.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Студент отримає знання і досвід розробки програм із застосуванням функціонального підходу. Оскільки інструменти функціонального програмування підтримуються багатьма сучасними мовами програмування, знання і розуміння функціонального програмування дасть змогу спростити архітектурні рішення, або ж надати альтернативи. Це також дасть змогу швидше опанувати програмні бібліотеки для різних мов програмування, що написані у функціональному стилі, або передбачають їхнє застосування у функціональному стилі.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Discipline	Functional programming in Common Lisp
Level of higher education	First (bachelor)
Course	4
Teaching hours	4 credits ECTS 54 год аудиторних занятъ, 66 год. СРС
Language of study	English
Department	System programming and specialized computer systems
Prerequisites	The discipline "Functional programming in Common Lisp" must be preceded by the disciplines "Data structures and algorithms", "Programming", "Object-oriented programming"
What will be studied	<ul style="list-style-type: none"> - basic principles of functional programming in Common Lisp programming language; - Common Lisp data structures and standard functions; - recursion and its application in functional programming; - the concept of higher-order functions, clojures, meta-object protocol, Common Lisp Object System and their application in Common Lisp; - the concept of macros as yet another approach to extend core language.
Why is it of interest / necessary to study	<ul style="list-style-type: none"> - Common Lisp is a multi-paradigmatic programming language with functional paradigm as the base one – it includes various concepts that are mutual to most modern programming languages; - mastering functional programming in Common Lisp allows a developer to expand their toolset and improve overall quality of the software they write; - Common Lisp itself as well as various Lisp dialects are used in multiple modern projects.
What you can learn (studying outcomes)	<ul style="list-style-type: none"> - write simple functional programs in Common Lisp; - implement complex programs in Common Lisp using multiple programming paradigms to solve typical problems in the field of computer and software engineering; - use standard higher-order functions and macros as well as implement new ones.
How to use the acquired knowledge and skills (competencies)	The student will gain knowledge and experience in developing programs using a functional approach in Common Lisp. Since functional programming paradigm elements are implemented in many modern programming languages, knowledge and understanding of functional programming in Common Lisp will help to implement state-of-the-art solutions or improve existing ones in such fields as compilers, static code analysis tools, big data, AI and others.
Information support	Training and working programs of the discipline, RSA, a set of educational and methodological materials
Form of classes	Lectures, lab classes
Semester control	Test

Discipline	Introduction to functional programming
Level of higher education	First (bachelor)
Course	4
Teaching hours	4 credits ECTS 54 год аудиторних занятъ, 66 год. СРС
Language of study	English
Department	System programming and specialized computer systems
Prerequisites	The discipline "Introduction to functional programming" must be preceded by the disciplines "Data structures and algorithms", "Programming"
What will be studied	<ul style="list-style-type: none"> - basic principles of functional programming, its advantages and disadvantages using the Common Lisp programming language as an example; - types of recursion and its application along with constructive approach for solving simple tasks; - approaches to working with recursive data structures such as single-linked lists; - the concept of higher-order functions and their application in programming.
Why is it of interest / necessary to study	<ul style="list-style-type: none"> - the functional programming paradigm is implemented to a some extent in most modern programming languages, which creates a need to study and understand it; - mastering functional programming allows a developer to expand their toolset and improve the quality of the software code they write; - the functional approach to programming allows one to formalize problems and consider them within the framework of a certain mathematical apparatus, which greatly simplifies the analysis of such problems.
What you can learn (studying outcomes)	<ul style="list-style-type: none"> - apply recursion and work with recursive data structures; - implement programs applying constructive approach; - work with functions as objects, including using typical higher-order functions such as map, filter, reduce, etc.
How to use the acquired knowledge and skills (competencies)	The student will gain knowledge and experience in developing programs using a functional approach. Since functional programming tools are supported by many modern programming languages, knowledge and understanding of functional programming will allow to simplify or provide alternative architectural solutions. It will also allow to more quickly master software libraries for various programming languages, that are written in a functional style or require functional style for application.
Information support	Training and working programs of the discipline, RSA, a set of educational and methodological materials
Form of classes	Lectures, lab classes
Semester control	Test

Дисципліна	Архітектура комп'ютера. Апаратне забезпечення
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Архітектура комп'ютера. Апаратне забезпечення» повинні передувати дисципліни "Архітектура комп'ютера. Програмне забезпечення" та "Системне програмне забезпечення"
Що буде вивчатися	<ul style="list-style-type: none"> - методи та способи організації взаємодії з апаратним забезпеченням на програмному рівні; - методи та способи оптимізації програмного коду з урахуванням можливостей апаратного забезпечення; - особливості організації продуктивних обчислень в гетерогенних розподілених системах.
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> - розглядаються особливості апаратної реалізації сучасних комп'ютерних систем в контексті їх ефективного використання, що може бути корисно при побудові високонавантажених програмних компонентів; - в рамках дисципліни усі програмні приклади подаються з використанням сучасної мови програмування Go, яка є на сьогодні стандартом в розробці високопродуктивних систем.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - максимально ефективно використовувати можливості апаратних компонентів та низькорівневої оптимізації; - створювати розподілені обчислювальні системи; - вирішувати задачі забезпечення консенсусу в розподілених системах.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Студент отримає досвід і зможе правильно використовувати засоби профілювання та низькорівневої оптимізації, а також засоби розробки та відлагоджування розподілених програмно-апаратних комплексів.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Архітектура комп'ютера. Архітектура вбудованих систем
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 54 год. аудиторної роботи, 66 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Архітектура комп'ютера. Апаратне забезпечення» повинні передувати дисципліни "Архітектура комп'ютера. Програмне забезпечення" та "Системне програмне забезпечення"
Що буде вивчатися	<ul style="list-style-type: none"> - основи архітектури вбудованих систем; - методи та способи організації програмного коду для ефективної взаємодії з вбудованими системами; - методи та способи профілювання та оптимізації програмних компонентів з урахуванням обмежень апаратного забезпечення.
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> - розглядаються особливості розробки окремих компонентів програмного забезпечення вбудованих систем, які можуть бути використані в сучасних роботизованих апаратних комплексах; - в рамках дисципліни розглядаються питання специфічних особливостей розробки програмного забезпечення вбудованих систем, які впливають на продуктивність.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - максимально ефективно використовувати можливості апаратних компонентів вбудованих систем; - використовувати протоколи обміну даними між апаратними компонентами.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Студент отримає досвід розробки програмного забезпечення для вбудованих систем, а також його профілювання та оптимізації.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Discipline	Computer architecture. Hardware
Level of higher education	First (bachelor's)
Course	4
Amount	4 credits 54 год. аудиторної роботи, 66 год. СРС
Language	English
Department	System programming and specialized computer systems
Prerequisites	Disciplines "Computer Architecture. Hardware" must be preceded by the disciplines "Computer Architecture. Software" and "System Software"
What will be studied	<ul style="list-style-type: none"> - methods of organizing interaction with hardware at the software level; - methods to optimize the software taking into account the capabilities of the hardware; - features of the organization of efficient computations in heterogeneous distributed systems.
Why it is interesting / necessary to study	<ul style="list-style-type: none"> - features of hardware implementation of modern computer systems in the context of their effective use are considered, which can be useful in building high-load software components; - within the discipline, all software examples are presented using the modern Go programming language, which is now the standard in the development of high-performance systems
What you can learn (learning outcomes)	<ul style="list-style-type: none"> - the most efficient using of the capabilities of hardware components and low-level optimization; - create distributed computer systems; - solve the problem of ensuring consensus in distributed systems.
How to use the acquired knowledge and skills (competencies)	The student will gain experience and will be able to properly use the profiling and low-level optimization tools, as well as tools for developing and debugging distributed software and hardware.
Information support	Training and working programs of the discipline, RSO, a set of educational and methodological materials
Form of classes	Lectures, labs
Semester control	Test

Дисципліна	Системне програмне забезпечення
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Системне програмне забезпечення» передують дисципліни «Архітектура комп'ютерів», «Комп'ютерна схемотехніка», «Комп'ютерні системи» навчального плану першого бакалаврського рівня вищої освіти;
Що буде вивчатися	<ul style="list-style-type: none"> - Архітектура системи Linux: Технічні вимоги; Контексти виконання в ядрі. - Віртуальна пам'ять. Обмеження ресурсів. Динамічне розподіл пам'яті. Проблеми з пам'яттю Linux. Debugging Tools для вирішення проблем з пам'яттю. - Обробка облікових даних. Традиційна модель дозволів Unix. - Можливості процесу. Сучасна модель можливостей POSIX. Виконання процесу. Перетворення програми в процес. Створення процесу. - Сигнальний механізм. Обробка процесу аварійного завершення. Сигнали в режимі реального часу. Таймери. Новіший механізм таймерів POSIX (інтервальний). - Багатопотоковість з Pthreads. Багатопотокові концепції. Управління потоками - найважливіші API pthread. Використання pthread API для синхронізації. Безпека pthread. Зниження та очищення pthread. - Планування процесора в Linux 686. ОС Linux та модель планування POSIX. - Можливості реального часу Linux. RTL - Linux як RTOS. - Розширений ввід-вивід файлів. Рекомендації щодо ефективності вводу-виводу. - Усунення несправностей та найкращі практики.
Чому це цікаво/треба вивчати	ОС Linux та її вбудовані та серверні програми є критично важливими компонентами сьогодношньої ключової програмної інфраструктури в децентралізованому та мережевому всесвіті. Попит у галузі на досвідчених розробників Linux постійно зростає. Цей курс надасть студенту дві речі: надійну теоретичну базу та практичну інформацію, що стосується галузі, ілюстровану кодом, що охоплює область системного програмування Linux. Цей курс поглиблює мистецтво та науку системного програмування Linux, включаючи архітектуру системи, віртуальну пам'ять та управління процесами, сигналізацію, таймери, багатопоточність, планування та введення / виведення файлів, основні підсистеми ядра Linux.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - практичному програмуванню системи з Linux, яке призначене для студентів, а також професіоналів Linux: системних інженерів, програмістів та тестувальників (QA); - використанню набору API, щоб зрозуміти теоретичні основи та концепції потужних API системного програмування Linux; - освоєнню розробки ядра Linux, його внутрішній дизайн, а також різні основні підсистеми. Ключовий код ядра, основні структури даних, функції та макроси, всебічна основа деталей реалізації основних служб та механізмів ядра.
Як можна користуватися набутими	Цей курс намагається вийти за межі використання API X для підходу Y; пояснити концепції та теорію, необхідні для розуміння інтерфейсів програмування, дизайнерських рішень та компромісів, зроблених досвідченими розробниками при

знаннями і уміннями (компетентності)	їх використанні та обґрунтування. Поради щодо усунення несправностей та найкращі галузеві практики доповнюють висвітлення курсу. Студент отримує концептуальні знання, а також досвід роботи, необхідні для роботи з інтерфейсами системного програмування Linux.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Програмне забезпечення розподілених операційних систем реального часу
Рівень ВО	Перший (бакалаврський)
Курс	
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Системне програмне забезпечення» передують дисципліни «Архітектура комп'ютерів», «Комп'ютерна схемотехніка», «Комп'ютерні системи» навчального плану першого бакалаврського рівня вищої освіти;
Що буде вивчатися	<ul style="list-style-type: none"> - Розподілені операційні системи реального часу і проміжне програмне забезпечення. Зв'язок в реальному часі. Трафік в реальному часі. Проблеми в розподілених системах реального часу з вбудованими елементами. - Загальні операційні системи та операційні системи реального часу. Управління завданнями. Синхронізація задач. Міжпроцесорна взаємодія в ОС Лінукс. Управління пам'яттю. Управління вводом/виводом. Огляд операційних систем реального часу. - Проєктування експериментального розподіленого ядра реального часу. Стратегія дизайну. Функції ядра. - Операційні системи реального часу. Проміжне програмне забезпечення реального часу. Синхронізація. Інтерфейс транспортного рівня. Вибір лідера в кільці. - Планування однопроцесорної незалежної задачі. Політики планування. Пріоритетне та неперіоритетне планування. Статичне, динамічне планування. Таксономія алгоритмів планування в реальному часі. Планування багатопроцесорних розподілених задач реального часу.
Чому це цікаво/треба вивчати	Розподілені системи реального часу з вбудованими елементами присутні всюди - від виробничих майданчиків заводів до автомобілів і авіоники. Розподілені системи реального часу характеризуються числом обчислювальних елементів, об'єднаних в мережу і виконують завдання в реальному часі. Завдання реального часу мають кінцеві терміни, і багато застосувань вимагають отримання рішення до їх закінчення. Сучасні технологічні досягнення призвели до значного збільшення числа елементів в розподілених системах реального часу, що, в свою чергу, викликало необхідність розробки відповідного програмного забезпечення.
Чому можна навчитися (результати навчання)	У цьому курсі можна навчитися розробці та впровадженню програмного забезпечення для розподілених систем реального часу з використанням методу висхідного проєктування. Як побудувати з нуля експериментальне ядро розподіленої операційної системи реального

	часу (experimental distributed real-time operating system kernel, DRTK) з розкриттям у процесі розробки всіх необхідних деталей.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробник або проєктувальник часто стикається з проблемою сполучення додатків з деякою комерційною операційною системою або проміжним програмним забезпеченням реального часу і часто змушений писати патчі до них. Це вимагає глибокого розуміння концепцій як апаратного забезпечення, так і операційної системи, що забезпечують обробку в реальному часі, і тому цей курс присвячений системному програмному забезпеченню операційних систем реального часу.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Розробка програмного забезпечення систем реального часу
Рівень ВО	Перший (бакалаврський)
Курс	
Обсяг	4 кредити ЄКТС 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліні «Системне програмне забезпечення» передують дисципліни «Архітектура комп'ютерів», «Комп'ютерна схемотехніка», «Комп'ютерні системи» навчального плану першого бакалаврського рівня вищої освіти;
Що буде вивчатися	<ul style="list-style-type: none"> - Життєвий цикл розробки програмного забезпечення: Модель Waterfall. V-модель. Спіральна модель Spiral Model. - Аналіз вимог і специфікація. Часовий аналіз. - Структурне проектування з діаграмами потоків даних. Об'єктно-орієнтоване проектування. - Методи реалізації в реальному часі: Кінцеві автомати, паралельні ієрархічні кінцеві автомати. - Тимчасові автомати. Мережі Петрі, тимчасові мережі Петрі. - Уніфікована мова моделювання в реальному часі: UML-діаграми. Функції реального часу. - Метод практичного проектування і реалізації. - Мови програмування в реальному часі: Додаток в реальному часі. - Операційна система C/POSIX в реальному часі: Інкапсуляція даних і управління модулями. Управління потоком POSIX (Управління часом. Синхронізація потоків і зв'язок. Сигнали. Взаємне виключення. Умовна синхронізація. Семафори.). - Обробка виключень і низькорівневе програмування. Реалізація управління процесом C/POSIX в реальному часі. - Fault Tolerance: Поняття і термінологія. Класифікація несправностей. Резервування. Відмовостійкі системи реального часу. - FaultTolerance в розподілених системах реального часу.
Чому це цікаво/треба вивчати	Системи реального часу присутні всюди, від автомобілів і мобільних телефонів до авіаційного обладнання та систем управління ядерними установками. Вони характеризуються своєчасністю реагування на деякий вхід. Помилка в реагуванні протягом певного часу може іноді привести до катастрофи. Правильність роботи системи реального часу залежить як від правильності результатів, так і від часу отримання цих результатів. Існує багато типів систем реального часу; система управління процесом в реальному часі отримує вхідні дані від датчиків, виконує деякі операції з цими даними і створює вихід для управління різними функціями системи, такими як включення і виключення, при необхідності активує аварійні сигнали і відображає системні дані.

Чому можна навчитися (результати навчання)	У цьому курсі можна навчитися розробці та впровадженню програмного забезпечення для систем реального часу. Тут розглядаються концепції зв'язку розподіленої операційної системи реального часу і розповідається, як спроектувати мережеві комунікації, щоб в режимі реального часу ядра могли взаємодіяти і мати розподілений системний програмний фрейм. Поступово в ході навчання, ви перетворите експериментальне ядро в ядро розподіленої операційної системи реального часу зі зв'язаним проміжним програмним забезпеченням, з'ясовуючи усі деталі реалізації.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Типовою є задача виконання всіх етапів розробки програмного забезпечення, починаючи з визначення вимог і розробки загального плану і до подальшої детальної розробки та кодування. Саме тут зустрічається більшість труднощів. В курсі надано метод для простого і ефективного виконання всіх етапів, а також проектування верхнього рівня і детальної розробки з використанням кінцевих автоматів, які можуть бути реалізовані із застосуванням потоків в операційній системі.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Комп'ютерний зір
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС, 72 год. аудиторної роботи, 48 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання мов програмування C++, Python, лінійної алгебри, мат. аналізу, алгоритмів та структур даних, чисельних методів обчислень
Що буде вивчатися	<ul style="list-style-type: none"> - Методи обробки та аналізу зображень - Методи сегментації та кластеризації зображень - Методи аналізу руху у відеопослідовностях - Процес калібрування камер, модель камери, внутрішні та зовнішні параметри камери та системи з декількох камер - Методи тривимірної реконструкції, геометрія декількох проекцій, принципи стереозору, ректифікація зображень з камер - Методи оцінки освітлення та затінення - Методи пошуку та відслідковування ключових точок на зображеннях - Методи виявлення об'єктів
Чому це цікаво/треба вивчати	Технології комп'ютерного зору дають змогу автоматизованим системам розв'язувати задачі виявлення, відстеження та ідентифікації об'єктів. Це актуальна галузь, що має широке практичне застосування, зокрема у системах відеоспостереження, автоматизації та підтримці виробничих процесів, медичних дослідженнях, автономних транспортних засобах, військових технологіях, створенні відеоефектів, а також у покращенні якості зображень.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - зрозуміти основні теоретичні засади методів комп'ютерного зору; - навчитися реалізовувати методи та алгоритми комп'ютерного зору за допомогою програмного коду; - ознайомитися із засобами бібліотеки OpenCV.
Як можна користуватися набутими знаннями і уміннями (компетентності)	отриманні знання будуть корисними при розробці, реалізації та супроводі програмних та програмно-апаратних засобів, пов'язаних із виявленням, відстежуванням та визначенням об'єктів
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, PCO, комплекс навчально-методичного забезпечення, конспект лекцій
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік

Дисципліна	Комп'ютерне забезпечення телекомунікацій
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 36 год. аудиторної роботи, 84 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліна базується на знаннях, отриманих студентами при вивченні таких дисциплін, як «Комп'ютерна електроніка», «Програмування», «Комп'ютерна схемотехніка», «Периферійні пристрої».
Що буде вивчатися	Курс дасть можливість орієнтуватися в світі телекомунікаційних технологій, що дозволить ефективно та швидко побудувати телекомунікаційну мережу будь якої складності з використанням останніх досягнень ведучих виробників активного та пасивного мережевого забезпечення.
Чому це цікаво/треба вивчати	Курс надасть знання, які дозволять аналізувати вимоги до телекомунікацій, обирати пасивні та активні елементи телекомунікаційної мережі відповідно до визначених вимог, що забезпечить побудову телекомунікаційної мережі будь якої складності.
Чому можна навчитися (результати навчання)	Методам побудови телекомунікаційної мережі, аналізу та моделюванню телекомунікаційної мережі, алгоритмів взаємодії елементів мережі та принципів побудови телекомунікаційної мережі з активним та пасивним обладнанням.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Після вивчення курсу студент набуде: уміння: – аналізувати схеми топологій телекомунікаційних мереж; – оцінювати складність побудови телекомунікаційної мережі; досвід: – проєктування телекомунікаційної мережі – налагодження апаратного та програмного забезпечення для активних компонентів мережі; – створення телекомунікаційної мережі згідно технічного завдання.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, методичні вказівки по виконанню лабораторних робіт, слайди лекцій
Форма проведення занять	Лекції та лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Комп'ютерна та телекомунікаційна техніка
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 36 год. аудиторної роботи, 84 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліна базується на знаннях, отриманих студентами при вивченні таких дисциплін, як «Комп'ютерна електроніка», «Програмування», «Комп'ютерна схемотехніка», «Периферійні пристрої».
Що буде вивчатися	Курс дасть можливість орієнтуватися в світі телекомунікаційної техніки. Будуть розглядатися: Сервери, що зберігають і обробляють інформацію; Робочі станції і ПК користувача; Активне обладнання — модеми, мережеві адаптери, концентратори, комутатори, маршрутизатори та ін. Мережеве програмне забезпечення.
Чому це цікаво/треба вивчати	Курс надасть знання, які дозволять створювати архітектуру комп'ютерної мережі з використанням як активного так і пасивного обладнання. Це допоможе спроектувати мережу будь якої складності під різні технічні вимоги.
Чому можна навчитися (результати навчання)	Методам побудови телекомунікаційної мережі, аналізу та моделюванню телекомунікаційної мережі, вибору телекомунікаційного обладнання для мережі згідно вимог.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Після вивчення курсу студент набуде: уміння: – аналізувати схеми топології телекомунікаційних мереж; – орієнтуватися в світі активного обладнання для побудови мережі; – оцінювати складність побудови телекомунікаційної мережі; досвід: – проектування телекомунікаційної мережі – налагодження апаратного та програмного забезпечення для активних компонентів мережі.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, методичні вказівки по виконанню лабораторних робіт, слайди лекцій
Форма проведення занять	Лекції та лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Телекомунікаційні технології
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 36 год. аудиторної роботи, 84 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Дисципліна базується на знаннях, отриманих студентами при вивченні таких дисциплін, як «Комп'ютерна електроніка», «Програмування», «Комп'ютерна схемотехніка», «Периферійні пристрої».
Що буде вивчатися	Принципи організації сучасних аналогових і цифрових систем, мереж зв'язку, включаючи комп'ютерні та Інтернет-мережі
Чому це цікаво/треба вивчати	Курс надасть знання, про організацію та побудову телекомунікаційних мереж різноманітного призначення. Будуть розглядатися методи організації телефонних, мобільних, телевізійних та комп'ютерних мереж, та додаткові сервіси що дають відповідні технології. Зокрема доступ до мережі інтернет .
Чому можна навчитися (результати навчання)	Технологіям організації мереж обміну інформації, як локальних так і глобальних. Принципам організації доступу до мережі використовуючи існуючі - як аналогові так і цифрові мережі. Будуть розглянуті апаратні засоби необхідні для підключення до існуючих мереж. Це дозволить будувати мережі з найменшими затратами, так як використовуються уже існуючі побудовані мережі.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Після вивчення курсу студент набуде: уміння: – Створювати мережі обміну інформацією на основі існуючих мереж; – Вибрати необхідне обладнання для підключення до мереж різних технологій; досвід: – проєктування телекомунікаційної мережі – налагодження апаратного та програмного забезпечення для активних компонентів мережі;
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, методичні вказівки по виконанню лабораторних робіт, слайди лекцій
Форма проведення занять	Лекції та лабораторні заняття
Семестровий контроль	Залік

Дисципліна	Управління ІТ-проектами
Рівень ВО	Перший (бакалаврський)
Курс	4 курс
Обсяг	4 кредити ECTS 36 год. аудиторної роботи, 84 год. СРС.
Мова викладання	українська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	Знання сучасних парадигм програмування, технологій проектування комп'ютерних систем, економіки та англійської мови
Що буде вивчатись?	Основні підходи та методи управління проектами в сфері ІТ, життєвий цикл проекту в ІТ, управління залученням стейкхолдерів, ризиками, змінами, бюджетом проекту, основи управління продуктом проекту.
Чому це цікаво/треба вивчати	Діяльність спеціалістів в сфері комп'ютерної інженерії нерозривно пов'язана з проектами, і саме тому для всебічного становлення спеціаліста необхідно здобути знання та розуміння засад, принципів та міжнародно визнаних підходів до управління проектами в сфері ІТ. Курс викладатиметься за сприяння та підтримки ГО "Інститут з управління проектами Україна" (PMI Ukraine Chapter).
Чому можна навчитися (результати навчання)	За умови успішного проходження курсу, студенти здобудуть такі знання: <ul style="list-style-type: none"> - основні підходи та методи управління ІТ проектами; - розуміння життєвого циклу ІТ проекту; - відмінності між проектом та продуктом проекту, та різницю в підходах до управління ними; - оцінювання доцільності використання конкретних підходів та методів управління в різноманітних проектних середовищах; - оцінювання термінів проектів; - засади та підходи до управління залученням, ризиками, змінами та бюджетом проекту; навички: - створення плану проекту та документації проекту; - навички створення та контролю бюджету проекту; - планування, кількісна та якісна оцінка ризиків; - підходи до організації та ведення проектної комунікації.
Як можна користуватися набутими знаннями та вміннями? (компетентності)	Після успішного проходження курсу, студент здобуде знання та навички, що сприятимуть успішній професійній та науковій діяльності, зокрема через вміння проаналізувати проектне середовище, вимоги, обмеження що постають перед виконанням великого обсягу робіт, скласти план проекту, визначити ризики та пропонувати план управління ризиками, оцінювати терміни та бюджет проекту, визначити доцільність застосування конкретних підходів та методів управління проектами і, як результат, досягати поставлених цілей проекту в зазначені терміни, з очікуваною якістю та в рамках бюджету проекту.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, конспект лекцій, презентації всіх лекцій, методичні рекомендації до виконання лабораторних робіт.
Форма проведення занять	Лекції та лабораторні роботи
Семестровий контроль	Залік

Дисципліна	Комп'ютерні системи штучного інтелекту
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредита ЄКТС 36 год. аудиторної роботи, 84 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання мов програмування С, С++, Пролог
Що буде вивчатися	<ul style="list-style-type: none"> - тенденції і перспективи розвитку систем штучного інтелекту; - принципи побудови та технологію розробки систем штучного інтелекту; синтаксис мови Пролог; основні стратегії розв'язування задач; - принципи побудови експертних систем; - принципи функціонування і способи застосування метаевристичних алгоритмів; - принципи побудови генетичних алгоритмів; - основні способи кодування хромосом в генетичних алгоритмах; - способи виконання генетичних операцій; - формулювання задачі штучного інтелекту в термінах декларативного програмування; - програмування задач штучного інтелекту засобами Пролога; - основні стратегії розв'язку задач; - механізми логічного виводу;
Чому це цікаво/треба вивчати	застосування парадигми штучного інтелекту принципово змінює погляди на підхід до проектування й реалізації програмних додатків
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - аналізувати вимоги до програми; - виконувати декомпозицію програми для забезпечення ефективної розробки й супроводження програми; - обирати й ефективно застосовувати стандартні бібліотеки й класи; - застосовувати засоби тестування й аналізу програм.
Як можна користуватися набутими знаннями і уміннями (компетентності)	отриманні знання будуть необхідними при розробці й супроводженні програм штучного інтелекту
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення, конспект лекцій
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік

Дисципліна	Штучні нейронні мережі
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредита ЄКТС 36 год. аудиторної роботи, 84 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання мов програмування С, С++, алгоритмів чисельних методів
Що буде вивчатися	<ul style="list-style-type: none"> - обчислювальна складність задач; комбінаторна оптимізація; - тенденції і перспективи розвитку систем штучного інтелекту; - принципи побудови нейронних мереж і підходи до навчання нейронних мереж; штучний нейрон, функції активації; - архітектури штучних нейронних мереж; - дельта правило; - алгоритм зворотного розповсюдження похибки; - карта Кохонена. Алгоритм навчання; - мережа Хопфілда.. Алгоритм навчання; - двоскерована асоціативна пам'ять; - мережа зустрічного поширення. Алгоритм навчання;
Чому це цікаво/треба вивчати	застосування парадигми штучного інтелекту принципово змінює погляди на підхід до проектування й реалізації програмних додатків
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - аналізувати вимоги до алгоритму оптимізації; - обирати архітектуру нейронної мережі відповідно до поставленої задачі; - обирати й ефективно застосовувати алгоритм навчання мережі; - застосовувати засоби тестування й аналізу програм.
Як можна користуватися набутими знаннями і уміннями (компетентності)	отриманні знання будуть необхідними при розробці й супроводженні програм штучного інтелекту
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення, конспект лекцій
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік

Дисципліна	Евристичні методи глобальної оптимізації
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредита ЄКТС 36 год. аудиторної роботи, 84 год. СРС
Мова викладання	Українська
Кафедра	системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання мов програмування С, С++, алгоритмів чисельних методів
Що буде вивчатися	<ul style="list-style-type: none"> - обчислювальна складність задач; - комбінаторна оптимізація; - тенденції і перспективи розвитку систем штучного інтелекту; - принципи функціонування й способи застосування метаевристичних алгоритмів; - алгоритм імітації відпалу; - алгоритм з заборонами; - алгоритм рою частинок; - алгоритм штучних світлячків; - алгоритм штучних кажанів; - алгоритм квіткового запилення; - застосування Levy Flight для модифікації ройових алгоритмів; - задача кластерізації, типи кластерізації; - методи оцінювання ефективності евристичних алгоритмів;
Чому це цікаво/треба вивчати	застосування парадигми штучного інтелекту принципово змінює погляди на підхід до проектування й реалізації програмних додатків
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> - аналізувати вимоги до алгоритму оптимізації; - обирати евристичний алгоритм відповідно до поставленої задачі; - оцінювати ефективність евристичних алгоритмів; - застосовувати засоби тестування й аналізу програм.
Як можна користуватися набутими знаннями і уміннями (компетентності)	отриманні знання будуть необхідними при розробці й супроводженні програм штучного інтелекту
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення, конспект лекцій
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік

Дисципліна	Computer systems of artificial intelligence
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредити ЄКТС 36 год. аудиторної роботи, 84 год. СРС
Мова викладання	Англійська
Кафедра	Системного програмування і спеціалізованих комп'ютерних систем
Вимоги до початку вивчення	знання в межах повної загальної середньої освіти, основ програмування та вищої математики
Що буде вивчатися	- базові положення в області нейронних мереж; - теоретичні положення створення та застосування глибоких нейронних мереж; - теорія та практика створення нейромережевих засобів, призначених для застосування в комп'ютерних системах.
Чому це цікаво/треба вивчати	отримані знання будуть необхідними та корисними для кожного фахівця в області комп'ютерної інженерії під час здійснення службових обов'язків, пов'язаних із застосуванням комп'ютерних систем штучного інтелекту.
Чому можна навчитися (результати навчання)	- знати базові засади теорії нейронних мереж; - знати технологію розробки та застосування нейромережевих засобів в комп'ютерних системах; - реалізовувати програмні механізми застосування нейромережевих засобів для вирішення практичних задач, пов'язаних з розпізнаванням графічної та текстової інформації.
Як можна користуватися набутими знаннями і уміннями (компетентності)	ФК 6. Здатність проектувати, впроваджувати та обслуговувати комп'ютерні системи та мережі різного виду та призначення. ФК 8. Здатність виконувати роботи з впровадження комп'ютерних систем та мереж, введення їх до експлуатації на об'єктах різного призначення. ФК 13. Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій. ФК 22. Здатність створювати програмне забезпечення та операційні системи нового нетрадиційного призначення.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, комплекс навчально-методичного забезпечення.
Форма проведення занять	Лекції, лабораторні заняття.
Семестровий контроль	Залік.