



**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ  
СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ  
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ  
СИСТЕМ**

**ЗАТВЕРДЖЕНО**

Методичною радою  
КПІ ім. Ігоря Сікорського  
(протокол №8 від «20» 06 2024 р.)

**Ф-КАТАЛОГ**

**ВИБІРКОВИХ НАВЧАЛЬНИХ ДИСЦИПЛІН  
ЦИКЛУ ПРОФЕСІЙНОЇ ПІДГОТОВКИ**

**для здобувачів ступеня магістр**

**за освітньою програмою «Інженерія програмного забезпечення мультимедійних та  
інформаційно-пошукових систем»**

**за спеціальністю 121 Інженерія програмного забезпечення**

**(вступ 2024 року)**

**УХВАЛЕНО:**

Вченою радою ФПМ  
КПІ ім. Ігоря Сікорського  
(протокол № 11 від «30» травня 2024 р.)

## Зміст

Інструкція користувачам каталогу	3
Ф-Каталог	4
<b>Анотації вибірових дисциплін для 1 курсу</b>	
1. Алгоритмічно-програмні методи захисту інформації	6
2. Algorithmic and Software Methods of Information Protection	8
3. Технології суперкомп'ютерних обчислень	10
4. High Performance Computing Technologies	12
5. Технології оброблення цифрових зображень та сигналів	14
6. Technologies of Processing Digital Images and Signals	16
7. Методи Block Chain	17
8. Block Chain Methods	18
9. Математичне моделювання систем і процесів	19
10. Mathematical Modeling of Systems and Processes	21
11. Проектування сучасних високопродуктивних обчислювальних систем	23
12. Design of Modern High-Performance Computer Systems	25
13. Машинне навчання	26
14. Machine Learning	28
<b>Анотації вибірових дисциплін для 2 курсу*</b>	
1. Проектування та розроблення мережевого програмного забезпечення	30
2. Network Software Design and Development	32
3. Технології Grid	34
4. Grid Technologies	35
5. Мережева архітектура та безпека IoT пристроїв	36
6. Network Architecture and IoT Device Security	38
7. Великі дані та аналітика	40
8. Big Data and Analytics	42
9. Креативні технології управління IT програмами та проектами	44
10. Creative Technologies for Managing IT Programs and Projects	46

\*) дисципліни вибору для освітньо-наукової програми («наукові» магістранти)

## Інструкція користувачам каталогу

1. Кількість і обсяг (у кредитах ЄКТС) навчальних дисциплін, які може обрати студент (вибіркових дисциплін) визначається навчальним планом, а саме

*для магістрів освітньо-професійної програми:*

1 курс – 23 кредити (2 дисципліни по 4 кредити, 3 дисципліни по 5 кредитів)

*для магістрів освітньо-наукової програми:*

1 курс – 23 кредити (2 дисципліни по 4 кредити, 3 дисципліни по 5 кредитів)

2 курс – 8 кредитів (2 дисципліни по 4 кредити).

2. Процедура вибору студентами навчальних дисциплін з Ф-каталогу реалізується через спеціалізовану інформаційну систему Університету [my.kpi.ua](http://my.kpi.ua) та включає такі етапи:

2.1 Реєстрація студента в [my.kpi.ua](http://my.kpi.ua).

2.2 Здійснення студентом вибору дисциплін.

2.3 Підтвердження студенту його вибору навчальних дисциплін із Ф-Каталогу.

2.4 Опрацювання результатів вибору дисциплін (фіксація результатів вибору) та передача даних для корекції індивідуальних навчальних планів студентів.

3. У разі неможливості формування повної академічної групи для вивчення певної вибіркової дисципліни, студентам надається можливість здійснити повторний вибір, приєднавшись до вже сформованих навчальних груп.

4. Студент не може двічі обрати одну й ту ж саму навчальну дисципліну. Студент не може обрати одну й ту ж саму навчальну дисципліну, яка викладається українською та англійською мовами одночасно (необхідно обрати або україномовний, або англійськомовний варіант).

5. Студент, який знехтував своїм правом вибору, буде записаний на вивчення тих дисциплін, які завідувач випускової кафедри вважатиме потрібними для оптимізації навчальних груп і потоків.

6. Якщо студент із поважної причини не зміг обрати дисципліни вчасно, або виявив помилку щодо свого волевиявлення, він звертається до деканату із заявою для запису на вивчення обраних ним дисциплін, надавши документів, які засвідчують поважність причин. Заява на зміну вибіркової дисципліни у сформованому індивідуальному навчальному плані має подаватися не пізніше ніж за місяць до початку семестру, в якому викладається ця дисципліна.

7. Не допускається зміна обраних дисциплін після початку навчального семестру, в якому вони викладаються.

6. Обрані студентом навчальні дисципліни зазначаються у його індивідуальному навчальному плані. Навчальні дисципліни, які внесені до індивідуального навчального плану здобувача, є обов'язковими для вивчення.

7. Більше інформації про порядок реалізації студентами права на вільний вибір навчальних дисциплін можна знайти у Положенні про реалізацію права на вільний вибір навчальних дисциплін здобувачами вищої освіти КПІ ім. Ігоря Сікорського.

## Ф-Каталог

### *Дисципліни для 1 курсу (вибір першокурсників)*

Потрібно обрати 23 кредити за рік: 2 дисципліни по 4 кредити, 3 дисципліни по 5 кредитів.

№	Назва навчальної дисципліни	Семестр	Кіл-ть кредитів	Семестрова атестація
1.	Алгоритмічно-програмні методи захисту інформації	2	4	залік
2.	Algorithmic and Software Methods of Information Protection	2	4	залік
3.	Технології суперкомп'ютерних обчислень	2	4	залік
4.	High Performance Computing Technologies	2	4	залік
5.	Технології оброблення цифрових зображень та сигналів	2	4	залік
6.	Technologies of Processing Digital Images and Signals	2	4	залік
7.	Методи Block Chain	2	5	екзамен
8.	Block Chain Methods	2	5	екзамен
9.	Математичне моделювання систем і процесів	2	5	екзамен
10.	Mathematical Modeling of Systems and Processes	2	5	екзамен
11.	Проектування сучасних високопродуктивних обчислювальних систем	2	5	екзамен
12.	Design of Modern High-Performance Computer Systems	2	5	екзамен
13.	Машинне навчання	2	5	екзамен
14.	Machine Learning	2	5	екзамен

**Дисципліни для 2 курсу (вибір першокурсників для освітньо-наукової програми)**

**Потрібно обрати 8 кредитів за рік: 2 дисципліни по 4 кредити.**

<b>№</b>	<b>Назва навчальної дисципліни</b>	<b>Семестр</b>	<b>Кіл-ть кредитів</b>	<b>Семестрова атестація</b>
1.	Проектування та розроблення мережевого програмного забезпечення	3	4	залік
2.	Network Software Design and Development	3	4	залік
3.	Технології Grid	3	4	залік
4.	Grid Technologies	3	4	залік
5.	Мережева архітектура та безпека IoT пристроїв	3	4	залік
6.	Network Architecture and IoT Device Security	3	4	залік
7.	Великі дані та аналітика	3	4	залік
8.	Big Data and Analytics	3	4	залік
9.	Креативні технології управління IT програмами та проектами	3	4	залік
10.	Creative Technologies for Managing IT Programs and Projects	3	4	залік

# Анотації вибірових дисциплін для 1 курсу

## 1. Алгоритмічно-програмні методи захисту інформації

Дисципліна	Алгоритмічно-програмні методи захисту інформації
Кафедра	Програмного забезпечення комп'ютерних систем
Рівень ВО	Другий (магістерський)
Курс, семестр	1 курс, 2 семестр
Обсяг дисципліни	4 кредити ЄКТС (лекції: 36 год., практ: 36 год., СРС: 66 год.)
Мова викладання	Українська
Вимоги до початку вивчення	Володіння знаннями із дисциплін "Програмне забезпечення систем автоматичної ідентифікації", "Моделювання та проектування інформаційних систем". Досвід командного розроблення програмного забезпечення.
Що буде вивчатися	Класифікація криптографічних систем, елементи криптоаналізу, сфери застосування криптографії, моноалфавітні шифри заміни, поліалфавітні шифри заміни, шифри перестановки, шифр Фейстеля, стандарт шифрування даних (DES), стандарт AES, міжнародний алгоритм шифрування даних (IDEA), Blowfish, RC5, арифметика у класах лишків за модулем, теорема Ферма та Ейлера, алгоритми перевірки чисел на простоту, арифметика довгих чисел, генерування випадкових простих чисел, принципи побудови криптосистем з відкритим ключем, обмін ключами за схемою Діффі-Хелмана, Алгоритм RSA та методи факторизації чисел, схема Ель-Гамала та методи знаходження дискретного логарифму, визначення еліптичної кривої, еліптичні криві над скінченними полями $GF(p)$ та $GF(p^m)$ , еліптичний аналог обміну ключами за схемою Діффі-Хелмана, алгоритми множення точки ЕК на число, задача дискретного логарифмування на еліптичній кривій, еліптичний аналог алгоритму RSA та схеми Ель-Гамала, принципи побудови криптостійких еліптичних кривих, стандарт цифрового підпису (DSS), алгоритм RSA та схема Ель-Гамала в режимі цифрового підпису, еліптичні алгоритми формування електронного цифрового підпису.
Чому це цікаво/треба вивчати	Метою кредитного модуля є формування у студентів здатностей аналізувати криптографічні системи, обирати криптографічний алгоритм відповідно до сформульованої задачі, забезпечувати роботу сучасних криптосистем, виконувати налагодження та розробку програмного забезпечення для криптографічного захисту.
Чому можна навчитися (результати навчання)	Знати: <ul style="list-style-type: none"><li>– методи захисту від несанкціонованого читання, контролю цілісності інформації, аутентифікації, захисту документів та цінних паперів від підробок;</li><li>– способи виконання операцій у скінченному полі виду <math>GF(p)</math> та його розширенні <math>GF(p^m)</math>, а також на еліптичній кривій;</li><li>– алгоритми розподілу ключів, формування цифрового підпису, шифрування даних;</li><li>– принципи розробки симетричних та асиметричних криптографічних алгоритмів;</li><li>– методики застосування сучасних криптографічних алгоритмів;</li></ul>

	<ul style="list-style-type: none"> <li>– особливості застосування криптографічних алгоритмів побудованих на еліптичній кривій.</li> </ul> <p>Уміти:</p> <ul style="list-style-type: none"> <li>– аналізувати результати роботи відомих криптографічних протоколів;</li> <li>– оцінювати стійкість криптографічних алгоритмів;</li> <li>– узагальнювати отримані експериментальні результати.</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	<p>Уміти:</p> <ul style="list-style-type: none"> <li>– проектувати системи криптографічного захисту;</li> <li>– розроблювати програмні засоби шифрування, дешифрування даних та створення електронного цифрового підпису;</li> <li>– створювати програмне забезпечення еліптичної криптографії;</li> <li>– застосовувати криптографічні стандарти при розробленні програмного забезпечення.</li> </ul>
<b>Інформаційне забезпечення</b>	<p>Силабус Електронний конспект лекцій</p>
<b>Семестровий контроль</b>	<p>Залік</p>

## 2. Algorithmic and Software Methods of Information Protection

Дисципліна	Algorithmic and Software Methods of Information Protection
<b>Кафедра</b>	Computer Systems Software Department
<b>Рівень ВО</b>	Second (Master)
<b>Курс, семестр</b>	1 <sup>st</sup> year, 2 <sup>nd</sup> semester
<b>Обсяг дисципліни</b>	4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours).
<b>Мова викладання</b>	English
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines "Software for Automatic Identification Systems", "Modeling and Design of Information Systems". Experience in team development of software.
<b>Що буде вивчатися</b>	Cryptographic systems classification. Cryptographic analysis elements. Cryptography application areas. Monoalphabetic replacement ciphers. Polyalphabetic replacement ciphers. Transposition cipher. Feistel cipher. Data encryption standard (DES). Advanced Encryption Standard (AES). International Data Encryption Algorithm (IDEA). Blowfish. Ron's Code 5 (RC5). Arithmetic in surplus classes by module. Fermat's and Euler's theorem. Algorithms of checking numbers on simplicity. Arithmetic of long numbers. Generate random prime numbers. Principles of building public-key cryptographic systems. Diffie–Hellman key exchange. RSA algorithm and number factorization methods. ElGamal scheme and methods of finding discrete logarithm. Elliptic curve determination. Elliptic curve over finite fields $GF(p)$ та $GF(p^m)$ . Elliptical analogue according to Diffie–Hellman key exchange scheme. Algorithms for multiplying point of elliptic curve on number. Discrete logarithm problem on an elliptic curve. Elliptical analog of RSA algorithm and ElGamal schemes. Principles of constructing crypto stable elliptic curves. Digital Signature Standard (DSS). RSA algorithm and ElGamal scheme in digital signature mode. Elliptical algorithms of forming electronic digital signature.
<b>Чому це цікаво/треба вивчати</b>	The aim of the course is to form students' ability to analyze cryptographic systems, select cryptographic algorithm according to formulated task, provision operation of modern cryptosystems, debug and develop software for cryptographic protection.
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– protection methods against unauthorized reading, control methods of information integrity, authentication methods, protection methods of documents and securities from forgeries;</li> <li>– methods of performing operations in finite field of form <math>GF(p)</math> and its extension <math>GF(p^m)</math>, as well as on elliptic curve;</li> <li>– algorithms of key distribution, digital signature generation, data encryption;</li> <li>– principles of developing symmetric and asymmetric cryptographic algorithm;</li> <li>– methods of applying modern cryptographic algorithms;</li> <li>– features of applying cryptographic algorithms based on elliptic curve.</li> </ul> Be able to: <ul style="list-style-type: none"> <li>– analyze work results of known cryptographic protocols;</li> </ul>



	<ul style="list-style-type: none"> <li>– estimate the stability of cryptographic algorithms;</li> <li>– summarize the obtained experimental results.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Be able to: <ul style="list-style-type: none"> <li>– design systems of cryptographic protection;</li> <li>– develop software of data encryption and decryption;</li> <li>– develop electronic digital signature;</li> <li>– develop software of elliptical cryptography;</li> <li>– apply cryptographic standards for software development.</li> </ul>
<b>Інформаційне забезпечення</b>	Syllabus Electronic lecture notes
<b>Семестровий контроль</b>	Credit

### 3. Технології суперкомп'ютерних обчислень

Дисципліна	Технології суперкомп'ютерних обчислень
Кафедра	Програмного забезпечення комп'ютерних систем
Рівень ВО	Другий (магістерський)
Курс, семестр	1 курс, 2 семестр
Обсяг дисципліни	4 кредити ЄКТС (лекції: 36 год., практ: 36 год., СРС: 66 год.)
Мова викладання	Українська
Вимоги до початку вивчення	Володіння знаннями з дисциплін "Основи програмування", "Програмування", "Бази даних".
Що буде вивчатися	Суперкомп'ютери та пов'язані з ними технології. Концепції та основи побудови високошвидкісних обчислювальних систем. Основні принципи та методи обчислень у високошвидкісних обчислювальних системах та комплексах різної архітектури. Основи архітектури багатопроцесорних обчислювальних систем. Архітектури та стандарти розподілених обчислень. Технології та принципи розподілених обчислень. Особливості та проблеми паралельної обробки та основні шляхи їх вирішення. Методи оцінки продуктивності швидкодіючих багатопроцесорних обчислювальних систем. Основні сценарії використання високошвидкісних обчислень.
Чому це цікаво/треба вивчати	Метою дисципліни є формування у студентів здатностей розробляти програмне забезпечення з максимальним використанням сучасного рівня існуючого технічного парку, застосовувати високошвидкісні обчислювальні системи для задоволення стрімкого зростання сучасних обчислювальних вимог.
Чому можна навчитися (результати навчання)	Уміти: <ul style="list-style-type: none"> <li>– створювати та використовувати високопродуктивні програмні системи на базі існуючих технічних ресурсів;</li> <li>– аналізувати вимоги до сучасних високопродуктивних комп'ютерних систем;</li> <li>– використовувати методи оцінки продуктивності обчислювальних систем;</li> <li>– розв'язувати задачі проєктування та експлуатації розподілених систем з підвищеними вимогами до обчислювальних можливостей;</li> <li>– розробляти нові архітектури високошвидкісних обчислювальних систем;</li> <li>– застосовувати теоретичні знання для розв'язання практичних задач підвищеної складності в плані безперервного зростання обчислювальних можливостей і швидкодії програмних систем;</li> <li>– освоїти сучасні технології Hyper-V.</li> </ul>
Як можна користуватися набутими знаннями і вміннями (компетентності)	Уміти: <ul style="list-style-type: none"> <li>– застосовувати технології суперкомп'ютерних обчислень;</li> <li>– використовувати на одному комп'ютері одночасно декілька операційних систем;</li> <li>– одночасно користуватись особливостями та програмами різних систем, не перенапружуючи комп'ютер;</li> </ul>

	<ul style="list-style-type: none"><li>– використовувати різні операційні системи для різних задач;</li><li>– розпаралелювати існуючі комп'ютерні ресурси.</li></ul>
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік

#### 4. High Performance Computing Technologies

<b>Дисципліна</b>	<b>High Performance Computing Technologies</b>
<b>Кафедра</b>	Computer Systems Software Department
<b>Рівень ВО</b>	Second (Master)
<b>Курс, семестр</b>	1 <sup>st</sup> year, 2 <sup>nd</sup> semester
<b>Обсяг дисципліни</b>	4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours).
<b>Мова викладання</b>	English
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines "Fundamentals of Programming", "Programming", "Databases".
<b>Що буде вивчатися</b>	Supercomputers and related software technologies. Concepts and bases of construction of high-speed computer systems. Basic principles and methods of calculations in high-speed computer systems and complexes of different architecture. Fundamentals of architecture of multiprocessor computer systems. Distributed computing architectures and standards. Technologies and principles of distributed computing. Features and problems of parallel processing and the main ways to solve them. Methods for evaluating the performance of high-speed multiprocessor computer systems. Basic scenarios for using high-speed computing.
<b>Чому це цікаво/треба вивчати</b>	The aim of the discipline is to develop students' ability to develop software with maximum usage of the current level of the existing technical park, to apply high-speed computer systems for meeting rapid growth of modern computing requirements.
<b>Чому можна навчитися (результати навчання)</b>	Be able to: <ul style="list-style-type: none"> <li>– create and use high-performance software systems based on existing technical resources;</li> <li>– analyze the requirements to modern high-performance computer systems;</li> <li>– use methods to assess the performance of computer systems;</li> <li>– solve problems of design and operation of distributed systems with increased requirements to computing capabilities;</li> <li>– develop new architectures of high-speed computer systems;</li> <li>– apply theoretical knowledge to solve practical problems of increased complexity in terms of continuous growth of computing power and performance of software systems;</li> <li>– apply modern Hyper-V technologies.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Be able to: <ul style="list-style-type: none"> <li>– apply technologies of supercomputer computing;</li> <li>– use several operating systems on one computer at the same time;</li> <li>– simultaneously use the features and programs of different systems without overloading the computer;</li> <li>– use different operating systems for different tasks;</li> <li>– parallelize existing computer resources.</li> </ul>
<b>Інформаційне забезпечення</b>	Syllabus Electronic lecture notes

<b>Семестровый контроль</b>	Credit
-----------------------------	--------

## 5. Технології оброблення цифрових зображень та сигналів

Дисципліна	Технології оброблення цифрових зображень та сигналів
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Рівень ВО</b>	Другий (магістерський)
<b>Курс, семестр</b>	1 курс, 2 семестр
<b>Обсяг дисципліни</b>	4 кредити ЄКТС (лекції: 36 год., практ: 36 год., СРС: 66 год.)
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Володіння знаннями з навчальних дисциплін «Технології штучного інтелекту для інформаційно-пошукових систем», «Моделювання та проєктування інформаційних систем». Досвід командного розроблення програмного забезпечення.
<b>Що буде вивчатися</b>	Інформаційні технології представлення цифрових зображень та сигналів. Технології розпізнавання зображень. Технології пошуку зображень. Технології перетворення даних для детектування об'єктів інтересу (Region of interest, ROI), трекінгу об'єктів інтересу (ROI tracing). Технології перетворення зображення для стиснення даних.
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є формування у студентів здатностей розробляти математичні та програмні інструменти дослідження зображень та сигналів, аналізувати інтелектуальні системи перетворення цифрових сигналів та зображень, обирати, налагоджувати та модифікувати технології аналізу сигналів відповідно до поставленої задачі.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– технології представлення цифрових зображень та сигналів;</li> <li>– технології класифікації зображень;</li> <li>– технології кластеризації зображень;</li> <li>– технології виділення ROI;</li> <li>– технології детектування ROI;</li> <li>– технології трекінгу ROI;</li> <li>– технології розпізнавання зображень.</li> </ul> Уміти: <ul style="list-style-type: none"> <li>– досліджувати сигнали та зображення з використанням відповідних інформаційних технологій;</li> <li>– аналізувати результати роботи програмних технологій розпізнавання зображень.</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– застосовувати методи, алгоритми, програмні засоби оброблення сигналів та зображень відповідно до поставлених завдань;</li> <li>– перетворювати методи, алгоритми, програмні засоби оброблення сигналів та зображень відповідно до поставлених завдань;</li> <li>– розроблювати математичні методи та програмні інструменти аналізу сигналів та зображень;</li> <li>– створювати інтелектуальні системи аналізу, розпізнавання та пошуку зображень.</li> </ul>

<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік

## 6. Technologies of Processing Digital Images and Signals

Дисципліна	Technologies of Processing Digital Images and Signals
<b>Кафедра</b>	Computer Systems Software Department
<b>Рівень ВО</b>	Second (Master)
<b>Курс, семестр</b>	1 <sup>st</sup> year, 2 <sup>nd</sup> semester
<b>Обсяг дисципліни</b>	4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours).
<b>Мова викладання</b>	English
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines "Artificial Intelligence Technologies for Information Retrieval Systems", "Information Systems Modeling and Design". Experience in team development of software.
<b>Що буде вивчатися</b>	Information technologies of representing the digital images and signals. Image recognition technologies. Image retrieval technologies. Data transformation technologies for detecting region of interest, tracing region of interest. Image transformation technologies for data compression.
<b>Чому це цікаво/треба вивчати</b>	The aim of the discipline is to form students' abilities to develop mathematical and software tools for researching images and signals, to analyze intelligent systems of transforming digital signals and images, to select, debug and transform technologies of signal analysis in according to the task.
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– technologies for representing digital images and signals;</li> <li>– image classification technologies;</li> <li>– image clustering technologies;</li> <li>– technologies for determining region of interest;</li> <li>– technologies for detecting region of interest;</li> <li>– technologies for tracing region of interest;</li> <li>– image recognition technologies.</li> </ul> Be able to: <ul style="list-style-type: none"> <li>– research signals and images using appropriate information technologies;</li> <li>– analyze results of software technologies for image recognition.</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Be able to: <ul style="list-style-type: none"> <li>– apply methods, algorithms, software tools for processing signals and images according to tasks;</li> <li>– transform methods, algorithms, software tools for processing signals and images according to tasks;</li> <li>– develop mathematical methods and software to analyze images and signals;</li> <li>– develop intelligent systems for image analysis, recognition and retrieval.</li> </ul>
<b>Інформаційне забезпечення</b>	Syllabus Electronic lecture notes
<b>Семестровий контроль</b>	Credit



## 7. Методи Block Chain

Дисципліна	Методи Block Chain
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Рівень ВО</b>	Другий (магістерський)
<b>Курс, семестр</b>	1 курс, 2 семестр
<b>Обсяг дисципліни</b>	5 кредитів ЄКТС (лекції: 36 год., практ: 36 год., СРС: 96 год.)
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Володіння знаннями з навчальних дисциплін «Організація комп'ютерних систем і мереж», «Основи програмування», «Програмування», «Бази даних», «Компоненти програмної інженерії».
<b>Що буде вивчатися</b>	Аналіз основного застосування технології блокчейн як грошової системи. Архітектура блокчейн-систем на прикладі системи Bitcoin. Криптографічні примітиви необхідні для функціонування блокчейн-системи. Використання технології блокчейн поза межами грошової системи. Експериментальні альтернативні протоколи (Ethereum, Monero), їхні переваги і недоліки.
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є формування у студентів здатностей розроблювати програмне забезпечення для взаємодії з мережею Bitcoin та розуміння моделі безпеки, яка є ключовим компонентом такого програмного забезпечення.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– переваги і недоліки використання технології блокчейн як грошової системи;</li> <li>– архітектуру програмного забезпечення за технологією блокчейн на прикладі системи Bitcoin;</li> <li>– основи криптографії;</li> <li>– поняття хеш-функцій та основи асиметричних криптосистем;</li> <li>– виділяти властивості блокчейн-систем, що дозволяють застосувати їх поза межами грошової системи;</li> <li>– розуміти переваги і недоліки альтернативних протоколів (Ethereum, Monero).</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– налаштовувати програмне забезпечення для взаємодії з системою Bitcoin;</li> <li>– розробляти прості програмні застосунки для взаємодії з системою Bitcoin;</li> <li>– аналізувати і адаптувати моделі безпеки програмних застосунків для взаємодії з блокчейн-системами.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Екзамен

## 8. Block Chain Methods

Дисципліна	Block Chain Methods
<b>Кафедра</b>	Computer Systems Software Department
<b>Рівень ВО</b>	Second (Master)
<b>Курс, семестр</b>	1 <sup>st</sup> year, 2 <sup>nd</sup> semester
<b>Обсяг дисципліни</b>	5 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 96 academic hours).
<b>Мова викладання</b>	English
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines "Fundamentals of Computer Systems and Networks", "Fundamentals of Programming", "Programming", "Databases", "Components of Software Engineering".
<b>Що буде вивчатися</b>	Analysis of the primary application of the blockchain technology as a monetary system. Architecture of the blockchain systems using Bitcoin system as a case study. Cryptographic primitives required for the functioning of the blockchain systems. Experimental alternative protocols (Ethereum, Monero), their advantages and drawbacks.
<b>Чому це цікаво/треба вивчати</b>	The aim of the discipline is to form students' abilities to develop software that can interact with Bitcoin network and understand the security model, which is the key component of the design of such software.
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– advantages and drawbacks of using blockchain technology as a monetary system;</li> <li>– architecture of blockchain systems and the software required to build such systems using Bitcoin network as a case study;</li> <li>– the notions of hash-functions and basics of asymmetric cryptosystems;</li> <li>– understand the properties of blockchain systems that make them useful outside the context of monetary systems;</li> <li>– understand advantages and drawbacks of alternative protocols (Ethereum and Monero).</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Be able to: <ul style="list-style-type: none"> <li>– setup and run software that interacts with Bitcoin network;</li> <li>– develop simple software applications for interacting with Bitcoin network;</li> <li>– analyze and adapt security models of software that interacts with blockchain systems.</li> </ul>
<b>Інформаційне забезпечення</b>	Syllabus Electronic lecture notes
<b>Семестровий контроль</b>	Exam

## 9. Математичне моделювання систем і процесів

Дисципліна	Математичне моделювання систем і процесів
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Рівень ВО</b>	Другий (магістерський)
<b>Курс, семестр</b>	1 курс, 2 семестр
<b>Обсяг дисципліни</b>	5 кредитів ЄКТС (лекції: 36 год., практ: 36 год., СРС: 96 год.)
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Володіння знаннями із дисциплін "Алгоритми та структури даних", "Програмування", "Лінійна алгебра та аналітична геометрія", "Математичний аналіз". Досвід роботи із математичним пакетом.
<b>Що буде вивчатися</b>	Класифікація моделей, властивості та контроль правильності математичних моделей. Елементи теорії похибок. Технологія моделювання. Математичні моделі, що описуються диференціальними рівняннями першого порядку. Математичні моделі, що описуються диференціальними рівняннями вищих порядків та їх системами. Аналіз систем диференціальних рівнянь та моделей, що описуються системами диференціальних рівнянь. Математичний пакет Matlab або Octave.
<b>Чому це цікаво/треба вивчати</b>	Обчислювальні (імітаційні) експерименти з моделями об'єктів дозволяють, спираючись на потужність сучасних обчислювальних методів та технічних інструментів інформатики, детально та глибоко вивчати об'єкти достатньо повно, що є недоступним для чисто теоретичного підходу. Математичне моделювання швидко розвивається, охоплюючи нові сфери – від розробки технічних систем та їх керуванням до аналізу складних економічних та соціальних процесів. Будучи методологією, математичне моделювання не замінює собою математику, фізику, біологію та інші наукові дисципліни і не конкурує з ними. Навпаки воно виконує синтезуючу роль.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– основи ймовірнісних методів аналізу і моделювання систем та методів імітаційного моделювання;</li> <li>– методи аналізу фазових портретів нелінійних систем диференціальних рівнянь;</li> <li>– загальні принципи побудови математичних моделей;</li> <li>– моделі зростання населення;</li> <li>– моделі конкуренції;</li> <li>– моделі механічних систем;</li> <li>– моделі взаємодії популяцій.</li> </ul> Уміти: <ul style="list-style-type: none"> <li>– будувати математичні моделі реальних систем та процесів;</li> <li>– проводити аналіз адекватності математичної моделі;</li> <li>– будувати та проводити багатокритеріальний аналіз моделей механічних систем, моделей зростання населення, конкуренції та взаємодії популяції;</li> <li>– обґрунтовувати вибір комп'ютерних засобів для побудови математичної моделі;</li> </ul>

	– застосовувати методи лінеаризації для спрощення моделей, що описуються нелінійними системами диференціальних рівнянь.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Уміти: – розроблювати математичні моделі реальних систем та процесів; – аналізувати адекватність математичної моделі; – застосовувати методи лінеаризації для спрощення моделей, що описуються нелінійними системами диференціальних рівнянь.
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Екзамен

## 10. Mathematical Modeling of Systems and Processes

Дисципліна	Mathematical Modeling of Systems and Processes
<b>Кафедра</b>	Computer Systems Software Department
<b>Рівень ВО</b>	Second (Master)
<b>Курс, семестр</b>	1 <sup>st</sup> year, 2 <sup>nd</sup> semester
<b>Обсяг дисципліни</b>	5 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 96 academic hours).
<b>Мова викладання</b>	English
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines "Algorithms and Data Structures", "Programming", "Linear Algebra and Analytic Geometry", "Higher Mathematics". Experience with a numerical computing environment.
<b>Що буде вивчатися</b>	Classification of models, properties and correctness control of mathematical models. Elements of error theory. Modeling technology. Mathematical models described by first-order differential equations. Mathematical models described by higher-order differential equations and their systems. Analysis of systems of differential equations and models described by differential systems equations. Numerical computing environment Matlab or Octave.
<b>Чому це цікаво/треба вивчати</b>	Computational (simulation) experiments with object models allow, based on the power of modern computational methods and technical tools of computer science, to study objects in detail and deeply enough, which is not available for a purely theoretical approach. Mathematical modeling is evolving rapidly, covering new areas – from the development of technical systems and their management to the analysis of complex economic and social processes. As a methodology, mathematical modeling does not replace mathematics, physics, biology and other scientific disciplines and does not compete with them. On the contrary, it plays a synthesizing role.
<b>Чому можна навчитися (результати навчання)</b>	<p>To know:</p> <ul style="list-style-type: none"> <li>– basics of probabilistic methods of analysis and modeling of systems and methods of simulation modeling;</li> <li>– methods of analysis of nonlinear systems of differential equations phase portraits;</li> <li>– general principles of building mathematical models;</li> <li>– population growth models;</li> <li>– competition models;</li> <li>– models of mechanical systems;</li> <li>– models of population interaction.</li> </ul> <p>Be able to:</p> <ul style="list-style-type: none"> <li>– build mathematical models of real systems and processes;</li> <li>– analyze the adequacy of the mathematical model;</li> <li>– build and conduct multi-criteria analysis of models of mechanical systems, models of population growth, competition and population interaction;</li> <li>– substantiate the choice of computer tools for building a mathematical model;</li> <li>– apply linearization methods to simplify the models described by nonlinear systems of differential equations.</li> </ul>

<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Be able to: <ul style="list-style-type: none"> <li>– develop of mathematical models of real systems and processes;</li> <li>– analyze of the adequacy of the mathematical model;</li> <li>– apply of linearization methods to simplify the models described by nonlinear systems of differential equations.</li> </ul>
<b>Інформаційне забезпечення</b>	Syllabus Electronic lecture notes
<b>Семестровий контроль</b>	Exam

11. Проектування сучасних високопродуктивних обчислювальних систем

Дисципліна	Проектування сучасних високопродуктивних обчислювальних систем
Кафедра	Програмного забезпечення комп'ютерних систем
Рівень ВО	Другий (магістерський)
Курс, семестр	1 курс, 2 семестр
Обсяг дисципліни	5 кредитів ЄКТС (лекції: 36 год., практ: 36 год., СРС: 96 год.)
Мова викладання	Українська
Вимоги до початку вивчення	Володіння знаннями із дисциплін "Основи програмування", "Програмування", "Бази даних".
Що буде вивчатися	Основи проектування високопродуктивних обчислювальних систем. Основні принципи та методи обчислень у високошвидких обчислювальних системах та комплексах різної архітектури. Технології, методи і засоби проектування високопродуктивних обчислювальних систем. Стандарти проектування та оформлення проектної документації. Інструментальні засоби проектування високопродуктивних систем. Технології Grid computing для побудови високопродуктивних обчислювальних систем. Технології Cloud computing для побудови високопродуктивних обчислювальних систем. Основні сценарії використання високошвидкісних обчислень.
Чому це цікаво/треба вивчати	Метою дисципліни є формування у студентів здатностей розробляти програмне забезпечення для проектування сучасних високопродуктивних обчислювальних систем, використовувати стороннє програмне забезпечення для створення сучасних високопродуктивних обчислювальних систем.
Чому можна навчитися (результати навчання)	<p>Уміти:</p> <ul style="list-style-type: none"> <li>– створювати високопродуктивні програмні системи на базі існуючих технічних ресурсів;</li> <li>– аналізувати вимоги до сучасних високопродуктивних комп'ютерних систем;</li> <li>– застосовувати технології Grid computing та Cloud computing для побудови високопродуктивних обчислювальних систем;</li> <li>– розв'язувати задачі проектування та експлуатації розподілених систем з підвищеними вимогами до обчислювальних можливостей;</li> <li>– розробляти нові архітектури високошвидких обчислювальних систем;</li> <li>– застосовувати теоретичні знання для розв'язання практичних задач підвищеної складності в плані безперервного зростання обчислювальних можливостей і швидкодії програмних систем.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Уміти:</p> <ul style="list-style-type: none"> <li>– застосовувати інструментальні засоби проектування сучасних високопродуктивних обчислювальних систем;</li> <li>– використовувати сучасні інтегровані технології для проектування високопродуктивних систем.</li> </ul>

<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Екзамен



## 12. Design of Modern High-Performance Computer Systems

Дисципліна	Design of Modern High-Performance Computer Systems
Кафедра	Computer Systems Software Department
Рівень ВО	Second (Master)
Курс, семестр	1 <sup>st</sup> year, 2 <sup>nd</sup> semester
Обсяг дисципліни	5 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 96 academic hours).
Мова викладання	English
Вимоги до початку вивчення	Knowledge of the disciplines "Fundamentals of Programming", "Programming", "Databases".
Що буде вивчатися	Fundamentals of designing high-performance computer systems. Basic principles and methods of calculations in high-speed computer systems and complexes of different architecture. Technologies, methods and tools for designing high-performance computer systems. Standards of design and execution of project documentation. High-performance system design tools. Grid computing technologies for building high-performance computing systems. Cloud computing technologies for building high-performance computing systems. Basic scenarios for using high-speed computing.
Чому це цікаво/треба вивчати	The aim of the discipline is to develop students' ability to design software to design modern high-performance computer systems, to use outside software for creating modern high-performance computer systems.
Чому можна навчитися (результати навчання)	Be able to: <ul style="list-style-type: none"> <li>– create and use high-performance software systems based on existing technical resources;</li> <li>– analyze the requirements to modern high-performance computer systems;</li> <li>– apply Grid and Cloud computing technologies to build high-performance computer systems;</li> <li>– solve problems of design and operation of distributed systems with increased requirements to computing capabilities;</li> <li>– develop new architectures of high-speed computer systems;</li> <li>– apply theoretical knowledge to solve practical problems of increased complexity in terms of continuous growth of computing power and performance of software systems.</li> </ul>
Як можна користуватися набутими знаннями і вміннями (компетентності)	Be able to: <ul style="list-style-type: none"> <li>– apply tools for designing modern high-performance computer systems;</li> <li>– use modern integrated technologies to design high-performance systems.</li> </ul>
Інформаційне забезпечення	Syllabus Electronic lecture notes
Семестровий контроль	Exam

### 13. Машинне навчання

Дисципліна	Машинне навчання
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Рівень ВО</b>	Другий (магістерський)
<b>Курс, семестр</b>	1 курс, 2 семестр
<b>Обсяг дисципліни</b>	5 кредитів ЄКТС (лекції: 36 год., практ: 36 год., СРС: 96 год.)
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Володіння знаннями із дисциплін «Теорія ймовірностей», «Основи програмування», «Програмування», «Бази даних», «Математичне забезпечення мультимедійних та інформаційно-пошукових систем», «Алгоритми та структури даних».
<b>Що буде вивчатися</b>	Методи класифікації та регресії. Метрики якості класифікації. Лінійні моделі класифікації та регресії, побудова та відбір ознак. Алгоритми побудови та навчання дерев рішень. Методи кластеризації. Способи поперечної підготовки даних. Нейронні мережі. Візуальний аналіз даних з Python. Мова програмування Python та її бібліотеки, зокрема numpy, pandas и scikit-learn.
<b>Чому це цікаво/треба вивчати</b>	Метою вивчення дисципліни є оволодіння навичками інтелектуального аналізу даних, аналітичного дослідження великих масивів даних з метою виявлення нових раніше невідомих знань, взаємозв'язків та закономірностей, опанування моделей, програмних продуктів та інструментальних засобів, які необхідні для проведення аналізу даних, вивчення типів задач, які вирішуються за допомогою машинного навчання, та прийомів розроблення та використання відповідного програмного забезпечення.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– основні поняття та визначення теорії машинного навчання;</li> <li>– основні моделі машинного навчання, що базуються на деревах прийняття рішень та нейронних мережах;</li> <li>– принципи побудови векторів ознак, вирішальних правил та класифікації;</li> <li>– основні види класифікаторів та принципи їх побудови;</li> <li>– методи кластеризації;</li> <li>– методи попереднього оброблення даних;</li> <li>– засоби формалізованого подання знань.</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– моделювати та аналізувати дані, маніпулювати ними;</li> <li>– обирати відповідний вид класифікатора в залежності від задачі, яка розв'язується;</li> <li>– проводити попередню обробку даних;</li> <li>– застосовувати алгоритми побудови і навчання класифікатора по вибірці;</li> <li>– виконувати обчислення, пов'язані з навчанням і роботою класифікатора;</li> </ul>

	<ul style="list-style-type: none"> <li>– розробляти структуру баз знань та баз даних;</li> <li>– розробляти алгоритми та програми для обробки баз даних та знань;</li> <li>– візуально відобразити результати проведених досліджень;</li> <li>– виконувати статистичне оцінювання.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Екзамен

## 14. Machine Learning

Дисципліна	Machine Learning
<b>Кафедра</b>	Computer Systems Software Department
<b>Рівень ВО</b>	Second (Master)
<b>Курс, семестр</b>	1 <sup>st</sup> year, 2 <sup>nd</sup> semester
<b>Обсяг дисципліни</b>	5 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 96 academic hours).
<b>Мова викладання</b>	English
<b>Вимоги до початку вивчення</b>	Knowledge in the disciplines "Probability Theory", "Fundamentals of Programming", "Programming", "Databases", "Mathematical Support of Multimedia and Information Retrieval Systems", "Algorithms and Data Structures".
<b>Що буде вивчатися</b>	Methods of classification and regression. Classification quality metrics. Linear models of classification and regression, construction and selection of features. Algorithms for constructing and learning decision trees. Clustering methods. Methods of preliminary data preparation. Neural networks. Visual data analysis using Python. Python programming language and its libraries, including numpy, pandas and scikit-learn.
<b>Чому це цікаво/треба вивчати</b>	The purpose of the discipline is to master the skills of data mining, analytical research of large data sets to identify new previously unknown knowledge, relationships and patterns, mastering models, software products and tools needed to analyze data, study the types of tasks that are solved using machine learning, and methods of development and use of the corresponding software.
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– basic concepts and definitions of machine learning theory;</li> <li>– basic models of machine learning based on decision trees and neural networks;</li> <li>– principles of features vectors construction, decisive rules and classification;</li> <li>– main types of classifiers and principles of their construction;</li> <li>– clustering methods;</li> <li>– methods of data pre-processing;</li> <li>– means of formalized knowledge presentation.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Be able to: <ul style="list-style-type: none"> <li>– model, analyze and manipulate data;</li> <li>– choose the appropriate type of classifier depending on the problem to be solved;</li> <li>– carry out preliminary data processing;</li> <li>– apply algorithms for constructing and training a classifier by sampling;</li> <li>– perform calculations related to the training and operation of the classifier;</li> <li>– develop the structure of knowledge bases and databases;</li> </ul>

	<ul style="list-style-type: none"><li>– develop algorithms and programs for databases and knowledge processing;</li><li>– perform statistical evaluation;</li><li>– display the results of research.</li></ul>
<b>Інформаційне забезпечення</b>	Syllabus Electronic lecture notes
<b>Семестровий контроль</b>	Exam

## Анотації вибіркових дисциплін для 2 курсу

### 1. Проектування та розроблення мережевого програмного забезпечення

Дисципліна	Проектування та розроблення мережевого програмного забезпечення
Кафедра	Програмного забезпечення комп'ютерних систем
Рівень ВО	Другий (магістерський)
Курс, семестр	2 курс, 3 семестр
Обсяг дисципліни	4 кредити ЄКТС (лекції: 36 год., практ: 36 год., СРС: 66 год.)
Мова викладання	Українська
Вимоги до початку вивчення	Знання дисциплін «Основи програмування», «Програмування», «Бази даних», «Основи комп'ютерних систем і мереж».
Що буде вивчатися	<p><b>Модуль 1</b></p> <p>Проектування програмного забезпечення для аналізу веб-сайтів, які надають відкриті дані мережі Інтернет. Використання бібліотек NumPy, Pandas, Matplotlib, Folium. Взаємодія програмного забезпечення з зовнішніми додатками. Pandas DataFrame. Призначення і можливості Speedtest-cli. Обчислення середньої швидкості Wi-Fi для набору даних. Імпорт даних з Інтернету за допомогою Pandas. Засоби Pandas для кореляційного аналізу. Регресійний аналіз даних мережі Інтернет. Розроблення програмного забезпечення для прогнозування трафіку мережі Інтернет засобами Python. Розроблення програмного забезпечення для аналітики та візуалізації даних мережі Інтернет. Виявлення евклідових аномалій Інтернет-трафіку. Програмне забезпечення для аналізу швидкості Інтернету за допомогою Scikit-Learn, Pandas та Matplotlib. Використання бібліотек Seaborn, FiveThirtyEight. Використання бібліотек Folium та Leaflet.js. Використання OpenStreetMap, Mapbox, Stamen для розроблення програмного забезпечення. Поєднання бази даних SQLite, файлів JSON та DataFrames. Імпорт скрипта Python з бібліотеки Folium, використання наборів Folium, модифікація та маркування карти. Lambda - архітектура обробки даних мережі Інтернет.</p> <p><b>Модуль 2</b></p> <p>Архітектура SDN. Призначення протоколу OpenFlow. APIC-EM. REST API. Структура запиту REST. Призначення і можливості Postman. Створення програм для запиту та відображення таблиці хостів та таблиці мережевих пристроїв у мережі. Інвентаризація хостів мережі за допомогою Python. Створення функції для запиту інвентаризації хоста. Аналіз Network Device Inventory API. Програмне забезпечення для введення даних користувача, читання та запису у зовнішні файли. Програмне забезпечення, яке отримує доступ до API на основі введення користувача та процесів відображення даних JSON. Парсинг JSON у Python. MapQuest API Application. Аутентифікація RESTful Request. Використання APIC-EM Path Trace API в Python. Використання DevNet GitHub. APIC-EM API для Flow Analysis. API Reference. The Rooms API. The List Messages API. List Messages Endpoint. List Messages Response. Використання Postman для надсилання запитів до ISS Location API. Створення виклику до API</p>

	Webex Teams в Python. Використання Python для Automate Listing Rooms. Використання Python для List and Search Messages. Використання Python для інтеграції API MapQuest і розташування ISS за допомогою API Webex.
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є формування у студентів здатностей проектувати та розроблювати програмне забезпечення для розв'язання практичних задач професійної діяльності, пов'язаних з аналізом відкритих даних мережі Інтернет.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– концепції мережевої програмованості та автоматизації з використанням технологій API, REST, JSON, Python, Git, Postman, SDN, NFV, NETCONF, RESTCONF, Controllers, Orchestration, YANG, YAML, ACI, APIC-EM та контейнерів.</li> </ul> Уміти: <ul style="list-style-type: none"> <li>– проектувати програмне забезпечення для збору та аналізу даних мережі Інтернет;</li> <li>– проектувати програмне забезпечення для взаємодії з API APIC-EM;</li> <li>– використовувати Python для здійснення викликів до API Webex Teams.</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– розроблювати програмне забезпечення, яке дозволить збирати та аналізувати відкриті дані мережі Інтернет;</li> <li>– розроблювати програмне забезпечення мереж SDN, включаючи центральний контроль за політикою прикладних програм;</li> <li>– використовувати DevNet Sandbox для взаємодії з програмованими пристроями за допомогою API реального світу на програмованих контролерах APIC-EM.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік

## 2. Network Software Design and Development

Дисципліна	Network Software Design and Development
Кафедра	Computer Systems Software Department
Рівень ВО	Second (Master)
Курс, семестр	2 <sup>nd</sup> year, 3 <sup>rd</sup> semester
Обсяг дисципліни	4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours).
Мова викладання	English
Вимоги до початку вивчення	Knowledge of the disciplines "Fundamentals of Programming", "Programming", "Databases", "Fundamentals of Computer Systems and Networks".
Що буде вивчатися	<p><b>Module 1</b></p> <p>Software design to analyze websites which provide open Internet data. Using libraries NumPy, Randas, Matplotlib, Folim. Software interaction with external applications. Pandas DataFrame. Purpose and capabilities of Speedtest-cli. Calculate the average Wi-Fi speed for data set. Import data from the Internet using Pandas. Pandas tools for correlation analysis. Regression analysis of Internet data. Software development for forecasting Internet traffic using Python. Software development for analytics and visualization of Internet data. Detection of Euclidean anomalies of Internet traffic. Internet speed analysis software with Scikit-Learn, Pandas and Matplotlib. Using libraries Seaborn, FiveThirtyEight. Using libraries Folium and Leaflet.js. Using OpenStreetMap, Mapbox, Stamen for software development. Using SQLite database, JSON files and DataFrames. Import Python script from Folium library, use Folium sets, modify and label the map. Lambda is Internet data processing architecture.</p> <p><b>Module 2</b></p> <p>SDN Architecture. OpenFlow protocol purpose. APIC-EM. REST API. REST query structure. Purpose and opportunities of Postman. Application development to query and display host tables and network device table on the network. Inventory of network hosts by means of Python. Create function to query host inventory. Analysis of Network Device Inventory API. Software for entering user data, reading and writing to external files. Software that accesses to API based on user input and processes of JSON data displays. JSON parsing in Python. MapQuest API Application. RESTful Request Authentication. Use APIC-EM Path Trace API in Python. Use DevNet GitHub. APIC-EM API for Flow Analysis. API Reference. The Rooms API. The List Messages API. List Messages Endpoint. List Messages Response. Use Postman for sending queries to ISS Location API. Create call to API Webex Teams in Python. Python usage for Automate Listing Rooms. Python usage for List and Search Messages. Python usage to integrate API MapQuest and ISS location by means of API Webex.</p>
Чому це цікаво/треба вивчати	The aim of the discipline is to develop students' ability to design and develop software to solve practical problems of professional activity related to the analysis of open data on the Internet.
Чому можна навчитися	To know:



<b>(результати навчання)</b>	<ul style="list-style-type: none"> <li>– concepts of network programming and automation using API, REST, JSON, Python, Git, Postman, SDN, NFV, NETCONF, RESTCONF, Controllers, Orchestration, YANG, YAML, ACI, APIC-EM and containers technologies.</li> </ul> <p>Be able to:</p> <ul style="list-style-type: none"> <li>– develop software to collect and analyse data on the Internet;</li> <li>– develop software for interaction with API APIC-EM;</li> <li>– use Python to implement calls to API Webex Teams.</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	<p>Be able to:</p> <ul style="list-style-type: none"> <li>– develop software which allows to collect and analyze open data on the Internet;</li> <li>– develop SDN network software including central control over application policy;</li> <li>– use DevNet Sandbox for interaction with programmable devices using the real-world API on APIC-EM programmable controllers.</li> </ul>
<b>Інформаційне забезпечення</b>	<p>Syllabus Electronic lecture notes</p>
<b>Семестровий контроль</b>	<p>Credit</p>

### 3. Технології Grid

Дисципліна	Технології Grid
Кафедра	Програмного забезпечення комп'ютерних систем
Рівень ВО	Другий (магістерський)
Курс, семестр	1 курс, 2 семестр
Обсяг дисципліни	4 кредити ЄКТС (лекції: 36 год., практ: 36 год., СРС: 66 год.)
Мова викладання	Українська
Вимоги до початку вивчення	Володіння знаннями з дисциплін спеціальності 121 «Інженерія програмного забезпечення», зокрема "Основи комп'ютерних систем і мереж", "Програмування" та інші. Досвід командного розроблення програмного забезпечення.
Що буде вивчатися	Grid технології та пов'язані з ними інформаційні технології. Концепції та основи Grid. Архітектури та стандарти Grid. Архітектури та механізми middleware Grid. Комунікаційні механізми Grid. Основні принципи та методи обчислень в Grid- системах. Особливості застосування Grid. Основні сценарії використання Grid.
Чому це цікаво/треба вивчати	Метою дисципліни є формування у студентів здатностей застосовувати системи Grid, middleware Grid та використання глобально розподілених обчислювальних систем для задоволення стрімкого зростання сучасних обчислювальних вимог.
Чому можна навчитися (результати навчання)	Уміти: <ul style="list-style-type: none"> <li>– працювати в Grid- системах;</li> <li>– використовувати високопродуктивні програмні системи на базі Grid;</li> <li>– аналізувати вимоги до сучасних високопродуктивних комп'ютерних систем;</li> <li>– розв'язувати обчислювальні задачі підвищеної складності за допомогою глобально розподілених систем.</li> </ul>
Як можна користуватися набутими знаннями і вміннями (компетентності)	Уміти: <ul style="list-style-type: none"> <li>– застосовувати технології високопродуктивних обчислень;</li> <li>– працювати в сучасних Grid-системах;</li> <li>– використовувати різні middleware Grid.</li> </ul>
Інформаційне забезпечення	Силабус Електронний конспект лекцій
Семестровий контроль	Залік

#### 4. Grid Technologies

<b>Дисципліна</b>	<b>Grid Technologies</b>
<b>Кафедра</b>	Computer Systems Software Department
<b>Рівень ВО</b>	Second (Master)
<b>Курс, семестр</b>	1 <sup>st</sup> year, 2 <sup>nd</sup> semester
<b>Обсяг дисципліни</b>	4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours).
<b>Мова викладання</b>	English
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines of specialty 121 "Software engineering", in particular "Fundamentals of computer systems and networks", "Programming" and others. Experience in team software development.
<b>Що буде вивчатися</b>	Grid technologies and related information technologies. Concepts and basics of Grid. Grid architectures and standards. Grid middleware architectures and mechanisms. Grid communication mechanisms. Basic principles and methods of calculations in Grid systems. Features of using Grid. Basic Grid usage scenarios.
<b>Чому це цікаво/треба вивчати</b>	The aim of the discipline is to form students' abilities to use Grid systems, Grid middleware and the use of globally distributed computing systems to meet the rapid growth of modern computing requirements.
<b>Чому можна навчитися (результати навчання)</b>	Be able to: <ul style="list-style-type: none"> <li>– work in Grid systems;</li> <li>– use high-performance software systems based on Grid;</li> <li>– analyze requirements for modern high-performance computer systems;</li> <li>– solve computational problems of increased complexity using globally distributed systems.</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Be able to: <ul style="list-style-type: none"> <li>– apply high-performance computing technologies;</li> <li>– work in modern Grid-systems;</li> <li>– use different Grid middleware.</li> </ul>
<b>Інформаційне забезпечення</b>	Syllabus Electronic lecture notes
<b>Семестровий контроль</b>	Credit

## 5. Мережева архітектура та безпека IoT пристроїв

Дисципліна	Мережева архітектура та безпека IoT пристроїв
Кафедра	Програмного забезпечення комп'ютерних систем
Рівень ВО	Другий (магістерський)
Курс, семестр	2 курс, 3 семестр
Обсяг дисципліни	4 кредити ЄКТС (лекції: 36 год., практ: 36 год., СРС: 66 год.)
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання дисциплін "Основи програмування", "Програмування", "Основи комп'ютерних систем і мереж".
Що буде вивчатися	<p><b>Модуль 1</b>                      Моделювання мережевої архітектури пристроїв IoT в середовищі Cisco Packet Tracer. Програмування Reg Server, MCU та SBC. Пристрої Wired IoT для Smart Home Network. Підключення до мережі та програмування пристроїв IoT. Додавання пристроїв IoT до Smart Home Network. Доступ та моніторинг мережі IoT. Підключення та моніторинг IoT-пристроїв. Використання Python для програмування пристроїв IoT. Налаштування віртуалізованого серверного середовища. Технологія прототипування. Використання фізичних матеріалів та електронних наборів інструментів для моделювання мережі пристроїв IoT. Налаштування PL-додатка за допомогою Raspberry Pi. Використання PL-App Notebook та Arduino для проектування мережі IoT.</p> <p><b>Модуль 2</b>                      Туманні обчислення в Smart Home Network. Хмара та хмарні обчислення. Розподілена обробка в мережі пристроїв IoT. Дослідження Smart Home Network. Машинне навчання в IoT. Створення прототипу програми AI. Intent-Based Networking (IBN). Use Cases для мереж IBN. Дослідження IBN. Загрози, вразливості та атаки мережі IoT. Кращі практики безпеки IoT пристроїв. Налаштування безпеки WPA2 на бездротовому маршрутизаторі. Налаштування VPN на смартфонах. Проектування безпечної веб-комунікації на веб-сервері в мережі Cloud Provider Service. Забезпечення цілісності даних пристроїв IoT.</p>
Чому це цікаво/треба вивчати	Метою дисципліни є формування у студентів здатностей проектувати і програмувати мережу пристроїв Інтернету речей, взаємодіяти з мережевими пристроями та гарантувати безпеку та цілісність даних в мережі пристроїв IoT.
Чому можна навчитися (результати навчання)	Уміти: <ul style="list-style-type: none"> <li>– застосовувати програмування для підтримки пристроїв IoT;</li> <li>– здійснювати проектування мереж пристроїв IoT;</li> <li>– забезпечувати цілісність даних пристроїв IoT;</li> <li>– налаштовувати безпеку мережевих пристроїв.</li> </ul>
Як можна користуватися набутими знаннями і вміннями (компетентності)	IoT створює попит на новий вид IT-спеціаліста. Це люди, які володіють знаннями та наборами навичок для розроблення нових продуктів з підтримкою IoT та обробки даних, які вони збирають. Потрібна підприємницька робоча сила, яка спеціалізується як на інформатиці, так і на програмному забезпеченні або комп'ютерній техніці.

	<p>На даний час є досить популярним і стрімко розвивається будівництво Smart Home Networks. Для налаштування такого типу мереж необхідні спеціалісти, які зможуть виконати програмування пристроїв IoT, налаштовувати мережеві пристрої для безпечної роботи пристроїв Smart Home Network без втрат даних та збереження їх цілісності.</p>
<b>Інформаційне забезпечення</b>	<p>Силабус Електронний конспект лекцій</p>
<b>Семестровий контроль</b>	<p>Залік</p>

## 6. Network Architecture and IoT Device Security

Дисципліна	Network Architecture and IoT Device Security
Кафедра	Computer Systems Software Department
Рівень ВО	Second (Master)
Курс, семестр	2 <sup>nd</sup> year, 3 <sup>rd</sup> semester
Обсяг дисципліни	4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours).
Мова викладання	English
Вимоги до початку вивчення	Knowledge of the disciplines "Fundamentals of Programming", "Programming", "Fundamentals of Computer Systems and Networks".
Що буде вивчатися	<p><b>Module 1</b></p> <p>Modelling network architecture of IoT devices in Cisco Packet Tracer environment. Reg Server, MCU and SBC programming. Wired IoT devices for Smart Home Network. Connection to network and IoT device programming. Addition of IoT devices to Smart Home Network. Access and monitoring of IoT network. Connection and monitoring of IoT devices. Usage Python to program IoT devices. Configuration of virtualized server environment. Prototype technology. Usage physical materials and electronic toolkits to model IoT device network. Configuration of PL-application by means of Raspberry Pi. Usage of PL-App Notebook and Arduino to design IoT network.</p> <p><b>Module 2</b></p> <p>Fog computing in Smart Home Network. Cloud and cloud computing. Distributed processing in IoT device network. Smart Home Network study. Machine learning in IoT. Creating prototype of AI program. Intent-Based Networking (IBN). Use Cases for IBN networks. IBN study. IoT network threats, vulnerabilities and attacks. Best practices of IoT device security. Configuration of WPA2 security on wireless router. Configuration of VPN on smartphones. Design of secure web communication on web server in the Cloud Provider Service network. Ensuring integrity of IoT device data.</p>
Чому це цікаво/треба вивчати	The aim of the discipline is to develop students' ability to design and program network of Internet devices, to interact with network devices and to ensure security and integrity of data in network of IoT devices.
Чому можна навчитися (результати навчання)	<p>Be able to:</p> <ul style="list-style-type: none"> <li>– apply programming to support IoT devices;</li> <li>– design network of IoT devices;</li> <li>– ensure integrity of IoT device data;</li> <li>– configurate security of network devices.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>IoT is creating demand for new kind of IT professional. These are people who have knowledge and skills to develop new IoT-enabled products and data processing which they collect. We need entrepreneurial workforce that specializes in both computer science and software.</p> <p>Currently, construction of Smart Home Networks is quite popular and rapidly developing. Configurating this type of network requires specialists who can perform programming of IoT devices, configure network devices for safe operation of Smart Home Network devices without data loss and preserve their integrity.</p>

<b>Інформаційне забезпечення</b>	Syllabus Electronic lecture notes
<b>Семестровий контроль</b>	Credit

## 7. Великі дані та аналітика

Дисципліна	Великі дані та аналітика
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Рівень ВО</b>	Другий (магістерський)
<b>Курс, семестр</b>	2 курс, 3 семестр
<b>Обсяг дисципліни</b>	4 кредити ЄКТС (лекції: 36 год., практ: 36 год., СРС: 66 год.)
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Знання дисциплін "Програмування", "Бази даних".
<b>Що буде вивчатися</b>	<p>Визначення та характеристики великих даних. Відкриті дані, їх формати та засоби обробки. Розроблення програмного забезпечення для аналізу вебсайтів, які надають відкриті дані з використанням мови Python. Призначення бібліотек NumPy, Randas, Matplotlib, Folim. Форматування даних про час та дату, читання та запис файлів за допомогою Python. Взаємодія з зовнішніми програмами. DataFrame. Інструменти Python для кореляційного аналізу. Регресійний аналіз даних мережі Інтернет за допомогою Python. Метод найменших квадратів. Програмування лінійної регресії за допомогою Python.</p> <p>Програмні методи кластеризації та асоціації даних у мережі. Аналіз основних компонентів. Аналіз незалежних компонентів. Алгоритми класифікації даних. Застосування та проблеми класифікацій. Модель класифікатора дерева рішень. Помилки в аналізі та прогнозуванні даних. Оцінка помилок регресії за допомогою Python. Модуль Pyplot. Аналіз швидкості Інтернету за допомогою Scikit-Learn, Pandas та Matplotlib. Архітектурні моделі інженерії великих даних. Технології віртуалізації. Гіпервізори. Контейнери. Контейнерна технологія для виконання програмного коду на сервері. SaaS, PaaS та IaaS. Технології Hadoop Big Data. Розподілена обробка MapReduce. HDFS. Поточкова платформа Kafka. Переваги Кафки. Переваги Cassandra та Hadoop для зберігання даних. Переваги Spark перед MapReduce. Лямбда - архітектура.</p> <p>Курс знайомить студентів із життєвим циклом аналізу даних. Студенти будуть використовувати блокноти Jupyter для збору, аналізу, візуалізації даних, представлення вмісту датасетів, рівнянь та запуску активного коду Python у межах одного файлу. Студенти також застосовуватимуть моделі машинного навчання для навчання комп'ютера для автоматизації завдань.</p>
<b>Чому це цікаво/треба вивчати</b>	<p>Дані є всюди. Важко зрозуміти, скільки нових даних генерується щодня. Даними можуть бути слова в книзі, вміст електронної таблиці, зображення, відео, аудіо, потік вимірювань, надіслані з пристрою моніторингу. Неочищені дані не мають корисного значення. Ми повинні обробити ці дані, а потім інтерпретувати результат, щоб зробити його корисним. Ці корисні дані тепер є інформацією. Коли ця інформація застосовується або розуміється, вона стає знанням. Коли даних настільки багато, що традиційні способи їх обробки, зберігання та аналізу не можуть бути використані, ми маємо справу з великими даними. Великі дані</p>



	вимагають нових методів та інструментів, щоб зробити їх значущими для досліджень.
<b>Чому можна навчитися (результати навчання)</b>	Цей курс знайомить студентів з технологіями та інструментами мови Python, які можна використовувати для аналітики великих даних. Студенти дізнаються, як аналізувати великі дані, використовуючи основні статистичні методи та технології оброблення даних у Python за допомогою Pandas, як аналізувати дані за допомогою моделей машинного навчання, як центри обробки даних та інженерія даних сприяють великим даним та аналітиці.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	З появою IoT та Big Data сьогодні з'являються нові категорії вакансій. Вчений з даних бере необроблені дані та перетворює їх на значущу інформацію. Вчені, що займаються даними, застосовують статистику, машинне навчання та аналітичні підходи для відповіді на критичні бізнес -питання. Вчені, що займаються даними, повинні інтерпретувати та передавати результати своїх висновків за допомогою методів візуалізації, створення додатків для аналізу даних. Вони працюють з наборами даних різного розміру та форм та запускають алгоритми на великих наборах даних. Вчені з даних повинні володіти найновішими технологіями. Вони повинні знати основи інформатики та програмування, включаючи досвід роботи з мовами та технологіями баз даних. Деякі інструменти, які допомагають дослідникам даних виконувати свою роботу, - це Python, R, Scala, Apache Spark, Hadoop, інструменти та алгоритми видобування даних, машинне навчання та статистика. Жодне з вищеописаних завдань не може існувати без інженера з даних. Інженери даних створюють інфраструктуру, яка підтримує Big Data. Вони розробляють та будують платформу, на якій усі ці дані зберігаються та обробляються. Інженери даних також керують усіма цими даними. Інженери даних можуть також інтегрувати дані з різних джерел і виконувати очищення даних. Деякі інструменти та програми, якими інженери даних регулярно користуються, це Hadoop, MapReduce, Hive, Pig, MySQL, MongoDB, Cassandra, Streaming Data, NoSQL, SQL та програмування.
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік

## 8. Big Data and Analytics

Дисципліна	Big Data and Analytics
Кафедра	Computer Systems Software Department
Рівень ВО	Second (Master)
Курс, семестр	2 <sup>nd</sup> year, 3 <sup>rd</sup> semester
Обсяг дисципліни	4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours).
Мова викладання	English
Вимоги до початку вивчення	Knowledge of the disciplines "Programming", "Databases".
Що буде вивчатися	<p>Definitions and characteristics of Big Data. Open data, their formats and processing means.</p> <p>Develop software to analyze websites that provide open data for using Python. Purpose of libraries NumPy, Randas, Matplotlib, Folim. Format time and date data, read and write files in Python. Interaction with external applications. DataFrame data. Purpose and capabilities of Speedtest-cli. Pandas tools for correlation analysis. Regression analysis of Internet data using Python. Least squares method. Linear regression programming using Python.</p> <p>Software methods of clustering and data association in the network. Analysis of the main components. Independent component analysis. Data classification algorithms. Application and problems of classifications. Decision tree classifier model. Errors in data analysis and forecasting. Estimation of regression errors by means of Python. Pyplot module. Plotly. Internet speed analysis using Scikit-Learn, Pandas and Matplotlib. Purpose of Seaborn, FiveThirtyEight libraries. Architectural models of Big Data engineering. Virtualization technologies. Hypervisors. Containers.</p> <p>Container technology for executing program code on the server. SaaS, PaaS and IaaS. Hadoop Big Data technologies. Distributed MapReduce processing. HDFS. Distributed Kafka streaming platform. Advantages of Kafka. Advantages of Cassandra vs Hadoop for data storage. Spark technology. Advantages of Spark vs MapReduce. Lambda - architecture.</p> <p>The course introduces the Data Analysis Lifecycle and the three types of data analytics: descriptive, predictive, and prescriptive. Students will use Jupyter Notebooks extensively to create a data pipeline to acquire, analyze, and visualize data. A Jupyter Notebook is a web application that allows students to be presented with content text, equations, visualizations, and to run live Python code within the same file. Students will also apply machine learning models to train a computer to automate tasks.</p>
Чому це цікаво/треба вивчати	<p>Data is everywhere. It can be hard to comprehend how much there is, and how much more new data is generated every day. Data can be the words in a book, the contents of a spreadsheet, pictures, video, audio, or a stream of measurements sent from a monitoring device. Raw data has little useful meaning. We must process this data and then interpret the output to make it useful. This useful data is now information. As this information is applied or understood, it becomes knowledge. When</p>

	<p>there is so much data that traditional ways of processing, storing and analyzing it cannot be used, that is called Big Data. Big Data requires new methods and tools to make it meaningful.</p>
<p><b>Чому можна навчитися (результати навчання)</b></p>	<p>This course introduces to those methods and tools to help you to harness the power of Big Data.</p> <p>Learn how to explain the promises and challenges of Big Data analytics as related to the Internet of Things.</p> <p>Learn how to analyze data using basic statistical and data preparation techniques in Python with Pandas.</p> <p>Learn how to analyze data using machine learning models.</p> <p>Learn different ways to defend a proposition using data, argumentation, and visualizations.</p> <p>Learn how to explain how data centers and data engineering contribute to Big Data and analytics.</p>
<p><b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b></p>	<p>With the emergence of the IoT and Big Data, there are new categories of jobs, and adjustments to existing jobs.</p> <p><i>Data Scientist.</i> A data scientist takes raw data and turns it into meaningful information. Data scientists apply statistics, machine learning and analytical approaches to answer critical business questions. Data science is an existing field that has expanded because of the IoT and Big Data. Like data analysts, data scientists must have data analytical skills. Data scientists must interpret and deliver the results of their findings, by visualization techniques, building data science apps, or narrating interesting stories about the solutions to their data (business) problems. They work with data sets of different sizes and shapes, and run algorithms on large data sets. Data scientists must be current with the latest technologies. They must know computer science fundamentals and programming, including experience with languages and database (big/small) technologies. Some of the tools that help data scientists do their work are Python, R, Scala, Apache Spark, Hadoop, data mining tools and algorithms, machine learning, and statistics.</p> <p><i>Data Engineer.</i> None of the three jobs detailed above can exist without the data engineer. The data engineer creates the infrastructure that supports Big Data. They design and build the platform on which all of this data is stored and processed. Data engineers also manage all this data. They ensure accessibility and availability for data scientists and data analysts.</p> <p>Data engineers may also integrate data from disparate sources and even perform some data cleaning. However, because data engineers primarily design the Big Data infrastructure for their company, they are not generally required to know any machine learning or analytics. Some of the tools and applications that data engineers routinely use are Hadoop, MapReduce, Hive, Pig, MySQL, MongoDB, Cassandra, Data streaming, NoSQL, SQL, and programming.</p>
<p><b>Інформаційне забезпечення</b></p>	<p>Syllabus Electronic lecture notes</p>
<p><b>Семестровий контроль</b></p>	<p>Credit</p>

## 9. Креативні технології управління ІТ програмами та проектами

Дисципліна	Креативні технології управління ІТ програмами та проектами
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Рівень ВО</b>	Другий (магістерський)
<b>Курс, семестр</b>	2 курс, 3 семестр
<b>Обсяг дисципліни</b>	4 кредити ЕКТС (лекції: 36 год., практ: 36 год., СРС: 66 год.)
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Володіння загальними знаннями із дисциплін комп'ютерної інженерії
<b>Що буде вивчатися</b>	Метою викладання дисципліни є засвоєння студентами креативних механізмів та технологій менеджменту інноваційних проектів та програм (P2M) в ІТ галузі. Формування практичних знань та вмінь з проактивного управління та впровадження інноваційних ІТ рішень. Набуття досвіду управління креативним потенціалом команди проекту. Набуття навичок роботи за міжнародним стандартом програмного та проектного менеджменту P2M.
<b>Чому це цікаво/треба вивчати</b>	Використання креативних технологій при створенні та впровадженні програмних продуктів є запорукою комерційного успіху на ринку ІТ. Цей курс дозволить набути фахових компетенцій з управління інноваційними розробками програмних продуктів та вмінь проактивного менеджменту, що сьогодні потребує міжнародний рівень спеціалістів з програмної інженерії.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– технології проактивного управління розробкою та впровадження програмного продукту;</li> <li>– технології трансферу знань в управлінні програмами та проектами;</li> <li>– технології креативного управління знаннями та комунікаціями в управлінні проектами та програмами;</li> <li>– формування стратегії розвитку програмного продукту при турбулентному оточенні;</li> <li>– принципів побудови програм в галузі ІТ;</li> <li>– особливостей застосування міжнародних стандартів менеджменту проектів та програм в галузі ІТ.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Цей курс рекомендується для слухачів, які планують мати кар'єрний ріст в проектах ІТ та працювати на рівні Senior Project Manager. Окрім того, цей курс дозволить працювати в міжнародних проектах з мінімальною адаптацією. Засвоєння навчальної дисципліни дозволить набути фахові компетенції: <ul style="list-style-type: none"> <li>– креативного та проактивного управління;</li> <li>– використовувати креативні технології при формуванні інноваційних ідей, при створенні інноваційних інформаційних технологій;</li> <li>– використовувати технології P2M для реалізації складних та комплексних проектів;</li> <li>– управляти знаннями та досвідом команди проекту;</li> <li>– використовувати креативний потенціал людських ресурсів.</li> </ul>

<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік

## 10. Creative Technologies for Managing IT Programs and Projects

Дисципліна	Creative Technologies for Managing IT Programs and Projects
<b>Кафедра</b>	Computer Systems Software Department
<b>Рівень ВО</b>	Second (Master)
<b>Курс, семестр</b>	2 <sup>nd</sup> year, 3 <sup>rd</sup> semester
<b>Обсяг дисципліни</b>	4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours).
<b>Мова викладання</b>	English
<b>Вимоги до початку вивчення</b>	General knowledge in the disciplines of computer engineering.
<b>Що буде вивчатися</b>	The aim of the discipline is assimilation by students the creative mechanisms and technologies of innovative projects and programs (P2M) management in the IT field. Formation of practical knowledge and skills in proactive management and implementation of innovative IT solutions. Gaining experience in managing the creative potential of the project team. Acquisition of skills to work according to the international standard of software and P2M project management.
<b>Чому це цікаво/треба вивчати</b>	The use of creative technologies in the creation and implementation of software products is the key to commercial success in the IT market. This course will allow students to acquire professional competencies in the management of innovative software development and proactive management skills, which today requires an international level of software engineering specialists.
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– technologies for proactive management of software product development and implementation;</li> <li>– technologies of knowledge transfer in program and project management;</li> <li>– technologies of creative management of knowledge and communications in project and program management;</li> <li>– formation of a software product development strategy in a turbulent environment;</li> <li>– principles of building programs in the field of IT;</li> <li>– features of application of international standards of management of projects and programs in the field of IT.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	This course is recommended for students who plan to have a career in IT projects and work at the Senior Project Manager level. In addition, this course will allow you to work in international projects with minimal adaptation. Mastering the discipline will allow you to acquire professional competencies of: <ul style="list-style-type: none"> <li>– creative and proactive management;</li> <li>– to use creative technologies in the formation of innovative ideas, in the creation of innovative information technologies;</li> <li>– to use P2M technologies for the implementation of complex and complex projects;</li> <li>– to manage the knowledge and experience of the project team;</li> </ul>

	– to use the creative potential of human resources.
<b>Інформаційне забезпечення</b>	Syllabus Electronic lecture notes
<b>Семестровий контроль</b>	Credit