

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ  
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

ЗАТВЕРДЖУЮ

Проректор з навчальної роботи

\_\_\_\_\_ Анатолій МЕЛЬНИЧЕНКО

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Ф-КАТАЛОГ**  
**ВИБІРКОВИХ НАВЧАЛЬНИХ ДИСЦИПЛІН**  
**ЦИКЛУ ПРОФЕСІЙНОЇ ПІДГОТОВКИ**  
для здобувачів ступеня бакалавра  
за освітньою програмою «Інженерія програмного забезпечення мультимедійних та  
інформаційно-пошукових систем»  
за спеціальністю 121 Інженерія програмного забезпечення  
(вступ 2019, 2020 року)

УХВАЛЕНО:

Методичною радою  
КПІ ім. Ігоря Сікорського  
(протокол №\_\_ від «\_\_» \_\_\_\_\_ 20\_\_ р.)

Вченою радою ФПМ  
КПІ ім. Ігоря Сікорського  
(протокол № 4 від «29» листопада 2021 р.)

## Зміст

Інструкція користувачам каталогу	4
Ф-Каталог	5
<b>Анотації вибіркових дисциплін для 3 курсу</b>	
1. Системне програмування	7
2. System Programming	9
3. Обчислювальні методи оптимізації	10
4. Computational Optimization Methods	12
5. Основи операційних систем	14
6. Operating System Fundamentals	16
7. Розроблення мобільних застосунків	18
8. Mobile Application Development	20
9. Розроблення вебзастосунків	21
10. Web-application Development	23
11. Теорія інформації та кодування	25
12. Information Theory and Coding	27
13. Управління стартап-проектами програмного забезпечення	29
14. Management of Software Startup Projects	31
15. Технології оброблення великих даних	33
16. Big Data Technologies	34
17. Абстрактна алгебра для інженерії програмного забезпечення	35
18. Abstract Algebra for Software Engineering	37
19. Кросплатформне програмування	39
20. Cross-Platform Programming	41
21. Адміністрування операційних систем Linux	43
22. Linux Operating Systems Administration	44
23. Менеджмент програмного продукту	45
24. Software Product Management	47
25. Web-технології та розробка web-систем	48
26. Прикладне програмне забезпечення мовою Python	49
27. Applied Data Science with Python	51
28. Аналіз даних	53
29. Data Analysis	55
<b>Анотації вибіркових дисциплін для 4 курсу</b>	
1. Програмні методи оброблення експериментальних даних	56
2. Software Methods of Experimental Data Processing	58
3. Бізнес-аналіз процесів автоматизації програмного забезпечення	59
4. Business Analysis of Software Automation Processes	61
5. Розроблення програмного забезпечення за методологією Agile	63
6. Agile Software Development	64
7. Технології DevOps	65
8. DevOps Technologies	67
9. Менеджмент проєктів ІТ	69
10. IT Project Management	71
11. Інфраструктура програмного забезпечення	73
12. Software Infrastructure	75
13. Програмні засоби захисту інформації	76
14. Software Instruments of Information Protection	78
15. Технологія Block Chain	80
16. Block Chain Technology	82
17. Емпіричні методи програмної інженерії	83

18. Empirical Software Engineering Methods	85
19. Основи хмарних технологій	87
20. Cloud Technologies Fundamentals	88
21. Методи побудови програмних засобів для матричних обчислень	89
22. Matrix Calculus Software Methods	91

## Інструкція користувачам каталогу

1. Кількість і обсяг (у кредитах ЄКТС) навчальних дисциплін, які може обрати студент (вибіркових дисциплін) визначається навчальним планом, а саме для 3 курсу – 32 кредити, для 4 курсу – 12 кредити. 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 ІС «my.kpi.ua».

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

7. 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.

## Ф-Каталог

### Дисципліни для 3 курсу (вибір другокурсниками)

Потрібно обрати 32 кредити за рік: 5-й семестр: 16 кредитів; 6-й семестр: 16 кредитів.

№	Назва навчальної дисципліни	Семестр	Кіл-ть кредитів	Семестрова атестація
1.	Системне програмування	5	4	залік
2.	System Programming	5	4	залік
3.	Обчислювальні методи оптимізації	5	4	залік
4.	Computational Optimization Methods	5	4	залік
5.	Основи операційних систем	5	4	залік
6.	Operating System Fundamentals	5	4	залік
7.	Розроблення мобільних застосунків	5	4	залік
8.	Mobile Application Development	5	4	залік
9.	Розроблення вебзастосунків	5	4	залік
10.	Web-application Development	5	4	залік
11.	Теорія інформації та кодування	5	4	залік
12.	Information Theory and Coding	5	4	залік
13.	Управління стартап-проектами програмного забезпечення	5	4	залік
14.	Management of Software Startup Projects	5	4	залік
15.	Технології оброблення великих даних	5	4	залік
16.	Big Data Technologies	5	4	залік
17.	Абстрактна алгебра для інженерії програмного забезпечення	6	4	залік
18.	Abstract Algebra for Software Engineering	6	4	залік
19.	Кросплатформне програмування	6	4	залік
20.	Cross-Platform Programming	6	4	залік
21.	Адміністрування операційних систем Linux	6	4	залік
22.	Linux Operating Systems Administration	6	4	залік
23.	Менеджмент програмного продукту	6	4	залік
24.	Software Product Management	6	4	залік
25.	Web-технології та розробка web-систем	6	4	залік
26.	Прикладне програмне забезпечення мовою Python	6	4	залік
27.	Applied Data Science with Python	6	4	залік
28.	Аналіз даних	6	4	залік
29.	Data Analysis	6	4	залік

### **Дисципліни для 4 курсу (вибір третьокурсниками)**

Потрібно обрати 12 кредитів за рік: 7-й семестр: 4 кредити; 8-й семестр: 8 кредитів.

<b>№</b>	<b>Назва навчальної дисципліни</b>	<b>Семестр</b>	<b>Кіл-ть кредитів</b>	<b>Семестрова атестація</b>
1.	Програмні методи оброблення експериментальних даних	7	4	залік
2.	Software Methods of Experimental Data Processing	7	4	залік
3.	Бізнес-аналіз процесів автоматизації програмного забезпечення	7	4	залік
4.	Business Analysis of Software Automation Processes	7	4	залік
5.	Розроблення програмного забезпечення за методологією Agile	7	4	залік
6.	Agile Software Development	7	4	залік
7.	Технології DevOps	7	4	залік
8.	DevOps Technologies	7	4	залік
9.	Менеджмент проектів ІТ	8	4	залік
10.	IT Project Management	8	4	залік
11.	Інфраструктура програмного забезпечення	8	4	залік
12.	Software Infrastructure	8	4	залік
13.	Програмні засоби захисту інформації	8	4	залік
14.	Software Instruments of Information Protection	8	4	залік
15.	Технологія Block Chain	8	4	залік
16.	Block Chain Technology	8	4	залік
17.	Емпіричні методи програмної інженерії	8	4	залік
18.	Empirical Software Engineering Methods	8	4	залік
19.	Основи хмарних технологій	8	4	залік
20.	Cloud Technologies Fundamentals	8	4	залік
21.	Методи побудови програмних засобів для матричних обчислень	8	4	залік
22.	Matrix Calculus Software Methods	8	4	залік

# Анотації вибірових дисциплін для 3 курсу

## 1. Системне програмування

Дисципліна	Системне програмування
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС
Мова викладання	Українська
Кафедра	Програмного забезпечення комп'ютерних систем
Вимоги до початку вивчення	<ul style="list-style-type: none"> <li>– володіння знаннями із дисциплін «Структури та алгоритми даних», «Основи програмування»;</li> <li>– досвід програмування мовою Сі (компілятор gcc).</li> </ul>
Що буде вивчатися	Програмні (API) та бінарні (ABI) інтерфейси операційних систем. Структура системних викликів на прикладі операційних систем стандарту POSIX. Методи керування процесами, виділенням пам'яті, методи взаємодії із сигналами, віртуальні файлові системи та керування їх об'єктами, структури фізичних файлових систем, основи системного часу та взаємодії із таймерами, методи міжпроцесної взаємодії, керування потоками, сокети та основи мережевого програмування.
Чому це цікаво/треба вивчати	Метою кредитного модуля є формування у студентів здатностей керувати апаратною платформою за допомогою програмного інтерфейсу операційної системи, аналізувати ресурси апаратної платформи та керувати їх розподілом між програмними процесами, оперувати потоками та розробляти мережеве програмне забезпечення, проектувати та розробляти системне програмне забезпечення відповідно до поставленого завдання.
Чому можна навчитися (результати навчання)	<p>Знання:</p> <ul style="list-style-type: none"> <li>– системного підходу до проектування програмного забезпечення автономних та вбудованих обчислювальних платформ;</li> <li>– сучасних методів проектування програмного забезпечення обчислювальних платформ у складі електронних пристроїв різного призначення;</li> <li>– ресурсів апаратних платформ та методів керування ними;</li> <li>– програмних інтерфейсів операційних систем.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Уміння:</p> <ul style="list-style-type: none"> <li>– користуватись програмними інтерфейсами операційних систем;</li> <li>– аналізувати розподіл ресурсів апаратної платформи та планувати їх використання;</li> <li>– розробляти сервісне програмне забезпечення та елементи операційних систем.</li> </ul>
Інформаційне забезпечення	Конспект лекцій Методичні вказівки до виконання лабораторних робіт
Форма проведення занять	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації

<b>Семестровий контроль</b>	Залік
-----------------------------	-------



## 2. System Programming

Дисципліна	System Programming
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 credits
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	<ul style="list-style-type: none"> <li>– knowledge of the disciplines "Algorithms and data structures", "Fundamentals of programming";</li> <li>– experience with C programming (gcc compiler);</li> <li>– knowledge of English.</li> </ul>
Що буде вивчатися	Software (API) and binary (ABI) interfaces of operating systems. Set of system calls of POSIX operating systems. Methods of process control, memory allocation, interaction with system signals, virtual file systems and management of its objects, structures of physical file systems, basics of system time and timers, methods of interprocess interaction, threading and thread control in multicore platforms, sockets and basics of network programming.
Чому це цікаво/треба вивчати	The aim of the course is to form students' ability to manage the hardware platform using the operating system software interface, analyze hardware platform resources and manage their distribution between processes, operate system threads and develop network software, design and develop system software.
Чому можна навчитися (результати навчання)	Knowledge of the: <ul style="list-style-type: none"> <li>– system approach to software design of autonomous and embedded computing platforms;</li> <li>– modern methods of designing software for computing platforms as part of electronic devices for various purposes;</li> <li>– hardware platform resources and management methods;</li> <li>– operating system software interfaces.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Skills: <ul style="list-style-type: none"> <li>– use software interfaces of operating systems;</li> <li>– analyze the distribution of hardware platform resources and plan their use;</li> <li>– develop service software and operating system elements.</li> </ul>
Інформаційне забезпечення	Each student has access to hands-on practice materials, labs, quizzes, and assessments to learn how to utilize the skills and knowledge gained on the course and interact with Linux OS.
Форма проведення занять	Lectures, computer classes, independent work of students, consultations
Семестровий контроль	Test

### 3. Обчислювальні методи оптимізації

Дисципліна	Обчислювальні методи оптимізації
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредитів ЄКТС
Мова викладання	Українська
Кафедра	Програмного забезпечення комп'ютерних систем
Вимоги до початку вивчення	<ul style="list-style-type: none"> <li>– володіння знаннями із дисциплін «Алгоритми та структури даних», «Програмування», «Вища математика»;</li> <li>– досвід розроблення програмного забезпечення на будь-якій мові програмування.</li> </ul>
Що буде вивчатися	Класифікація методів та алгоритмів оптимізації. Обчислювальна складність задач оптимізації. Клас алгоритмів детермінованого локального пошуку. Клас алгоритмів стохастичного локального пошуку. Генетичні алгоритми. Мурашині методи та алгоритми.
Чому це цікаво/треба вивчати	Стрімке розширення сфер застосування методів математичної оптимізації, що відбувається в останні роки, формує потребу в розв'язуванні нових класів задач підвищеної складності та розмірності. Побудова нових методів вимагає знання основ обчислювальних методів оптимізації, комп'ютерних технологій розв'язання практичних обчислювальних задач, кількісного обґрунтування технічних і управлінських рішень на основі використання методів оптимізації, одержання необхідної підготовки для застосування набутих знань при вивченні інших дисциплін.
Чому можна навчитися (результати навчання)	<p>Знати:</p> <ul style="list-style-type: none"> <li>– методи багатокритеріальної оптимізації;</li> <li>– еволютивні методи оптимізації;</li> <li>– стохастичні алгоритми знаходження екстремального значення функції.</li> </ul> <p>Уміти:</p> <ul style="list-style-type: none"> <li>– застосовувати на практиці методи багатокритеріальної оптимізації для побудови програмних систем;</li> <li>– застосовувати на практиці мурашині алгоритми для розроблення програмного забезпечення розв'язання математичних задач;</li> <li>– розробляти програмне забезпечення для реалізації стохастичних алгоритмів обчислення екстремального значення функції.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розроблення та ефективного застосування програмних додатків для обчислення екстремальних значень функцій однієї змінної та функції кількох змінних.
Інформаційне забезпечення	Завдання на лабораторні роботи

<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

#### 4. Computational Optimization Methods

Дисципліна	Computational Optimization Methods
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 credit
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	<ul style="list-style-type: none"> <li>– knowledge of the disciplines "Data Structures and Algorithms", "Programming", "Higher Mathematics";</li> <li>– experience in software development for any programming language.</li> </ul>
Що буде вивчатися	Classification of optimization methods and algorithms. Computational complexity of optimization tasks. Algorithms class of deterministic local search. Algorithms class of stochastic local search. Genetic algorithms. Ant methods and algorithms.
Чому це цікаво/треба вивчати	<p>Rapid expansion of application scope of mathematical optimization methods forms requirement for solving new classes of increased complexity and dimension tasks in recent years.</p> <p>Building new methods requires basic knowledge of computational optimization methods, computer technologies of solving practical computational tasks, quantitative substantiation of technical and managerial decisions based on using optimization methods.</p> <p>Obtainment of required training for applying acquired knowledge to study other disciplines.</p>
Чому можна навчитися (результати навчання)	<p>To know:</p> <ul style="list-style-type: none"> <li>– methods of multicriteria optimization;</li> <li>– evolutionary optimization methods;</li> <li>– stochastic algorithms for searching extreme value of function.</li> </ul> <p>Be able to:</p> <ul style="list-style-type: none"> <li>– apply methods of multicriteria optimization to develop program systems in practice;</li> <li>– apply ant algorithms to develop software for solving mathematical tasks in practice;</li> <li>– develop software for implementing stochastic algorithms of computing extreme value of function.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Development and effective application of software applications to compute extreme values of single-variable functions and several-variables functions.
Інформаційне забезпечення	Methodical instructions for laboratory work

<b>Форма проведения занятий</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровый контроль</b>	Test

## 5. Основи операційних систем

Дисципліна	Operating System Fundamentals
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями із дисциплін "Алгоритми та структури даних", "Основи програмування". Досвід роботи із ОС Linux та Windows на рівні користувача (GUI).
<b>Що буде вивчатися</b>	Керування віртуальними машинами. Структура та архітектура операційних систем (ОС), призначення, алгоритми та реалізація підсистем керування файловою системою, процесами, пам'яттю та мережею, конфігурування ОС та процес завантаження. Віддалений доступ SSH. Системні утиліти ОС Windows та Linux, Virtual Box, SSH.
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є вивчення особливостей основних компонентів ОС для розуміння внутрішньої будови та ефективного використання ОС. Набуті знання дозволять користуватися можливостями операційних систем Windows та Linux у повсякденній і професійній діяльності, усувати найпоширеніші проблеми у роботі локальних та віддалених операційних систем.
<b>Чому можна навчитися (результати навчання)</b>	Знання: <ul style="list-style-type: none"> <li>– призначення, особливості та керування віртуальними машинами;</li> <li>– призначення, архітектура, класифікація, апаратна підтримка, інтерфейси ОС;</li> <li>– архітектура, логічна та фізична реалізації файлових систем;</li> <li>– MBR та GPT структура розділів диску, процес форматування та монтування розділу, принципи резервного копіювання та архівації;</li> <li>– властивості файлових систем;</li> <li>– процес завантаження ОС, BIOS, UEFI, проблеми та їх вирішення;</li> <li>– особливості реалізації та керування процесами, сервісами, алгоритми планування;</li> <li>– керування пам'яттю, віртуальна пам'ять, підкачка, відповідні алгоритми;</li> <li>– модель OSI, MAC-address, ARP-table, MTU, IPv4, IPv6, CIDR, TCP, UDP, NAT, DNS, DHCP, http, https, ssh, ftp, Samba, сокети, порти, коди відповіді http;</li> <li>– призначення, види хмарних сервісів.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміння: <ul style="list-style-type: none"> <li>– керувати віртуальними машинами;</li> <li>– інстальювати, налаштовувати та супроводжувати операційні системи;</li> <li>– керувати, оцінювати, аналізувати та вирішувати проблеми із основними складовими ОС (файловою системою, процесами, пам'яттю, мережею);</li> <li>– налаштовувати та керувати процесом завантаження ОС;</li> </ul>

	<ul style="list-style-type: none"> <li>– ефективно здійснювати вибір файлових систем;</li> <li>– працювати в командному рядку;</li> <li>– обслуговувати локальні та віддалені ОС.</li> </ul>
<b>Інформаційне забезпечення</b>	<p>Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт</p>
<b>Форма проведення занять</b>	<p>Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації</p>
<b>Семестровий контроль</b>	<p>Залік</p>

## 6. Operating System Fundamentals

Дисципліна	Operating System Fundamentals
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 credits
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer Systems Software Department
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines "Algorithms and data structures", "Fundamentals of programming". Experience with Linux and Windows operating systems at the user level (GUI).
<b>Що буде вивчатися</b>	Structure and architecture of operating systems, purpose, algorithms and realization of file system management subsystems, processes, memory and network, configuring of operating system and loading process, virtualization technologies and cloud technologies. Windows and Linux system utilities: virtual machine management, monitoring of OS subsystems, remote access.
<b>Чому це цікаво/треба вивчати</b>	The aim of the discipline is to study features of main components of operating system and component control algorithms for understanding internal structure and effective using of operating system. Obtained knowledge will allow you to use capabilities of Windows and Linux operating systems in routine and professional activities and to eliminate the most common problems in the work of local and remote operating systems.
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– purpose, architecture, classification, hardware support, interfaces of operating systems;</li> <li>– architecture, logical and physical realizations of file systems;</li> <li>– MBR and GPT disk partition structures, processes of formatting and mounting partition, backup and archiving principles;</li> <li>– file system properties;</li> <li>– loading process of operating system, BIOS, UEFI, problems and their solutions;</li> <li>– features of implementing and managing processes, services, planning algorithms;</li> <li>– memory management, virtual memory, paging, appropriate algorithms;</li> <li>– model OSI, MAC-address, ARP-table, MTU, IPv4, IPv6, CIDR, TCP, UDP, NAT, DNS, DHCP, http, https, ssh, ftp, Samba, sockets, ports, http response codes;</li> <li>– purpose, features and management of virtual machines;</li> <li>– purpose, cloud service types.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Be able to: <ul style="list-style-type: none"> <li>– manage virtual machines;</li> <li>– install, configure and maintain operating systems;</li> <li>– manage, estimate, analyze and decide problems with main components of operating system (file system, processes, memory, network);</li> <li>– configure and manage process of loading operating system;</li> <li>– effectively select file systems;</li> </ul>



	– support local and remote operating systems.
<b>Інформаційне забезпечення</b>	Lecture notes Presentations Tutorials for laboratory works
<b>Форма проведення занять</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровий контроль</b>	Test

## 7. Розроблення мобільних застосунків

Дисципліна	Розроблення мобільних застосунків
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями з навчальних дисциплін «Операційні системи», «Основи програмування», «Об'єктно-орієнтоване програмування», «Бази даних».
<b>Що буде вивчатися</b>	Мобільні програмні застосунки. Мобільні операційні системи (Android, iOS). Мови програмування мобільних програмних застосунків (Java, Swift). Життєвий цикл мобільних програмних застосунків. Специфікація вимог до мобільних програмних застосунків. Статична та динамічна логічні структури мобільних програмних застосунків. Фізична структура мобільних програмних застосунків. Програмне реалізування мобільних застосунків за їх структурами. Мобільні операційні системи Android, iOS; мови програмування Java, Swift.
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є набуття теоретичних знань і практичних навиків розроблення мобільних програмних застосунків. Використання набутих практичних навиків дозволить проектувати, розробляти та експлуатувати мобільні програмні застосунки на прикладі операційних систем Android, iOS.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– особливості розроблення мобільних програмних застосунків;</li> <li>– життєвий цикл мобільного програмного забезпечення;</li> <li>– основи програмування мовами Java, Swift;</li> <li>– основи проектування мобільних програмних застосунків;</li> <li>– основи розроблення мобільних програмних застосунків;</li> <li>– основи тестування мобільних програмних застосунків;</li> <li>– особливості експлуатування мобільних програмних застосунків.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– специфікувати вимоги до мобільних програмних застосунків;</li> <li>– створювати статичну логічну структуру мобільних програмних застосунків;</li> <li>– створювати динамічну логічну структуру мобільних програмних застосунків;</li> <li>– створювати фізичну структуру мобільних програмних застосунків;</li> <li>– програмно реалізовувати мобільні програмні застосунки за їх структурами.</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт Методичні вказівки до самостійної робіт

<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 8. Mobile Application Development

Дисципліна	Mobile Application Development
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 credits
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	Knowledge of the disciplines "Operating systems", "Programming basics", "Object-oriented programming", "Database".
Що буде вивчатися	Mobile software applications. Mobile operating systems (Android, iOS). Programming languages of mobile software applications (Java, Swift). Life cycle of mobile software applications. Requirements specification to mobile software applications. Static and dynamic logical structures of mobile software applications. Physical structure of mobile software applications. Software implementation of mobile applications according to their structures. Android, iOS mobile operating systems. Java, Swift programming languages.
Чому це цікаво/треба вивчати	The purpose of discipline is to acquire theoretical knowledge and practical skills of mobile software application development. Using acquired practical skills will allow you to design, develop and operate mobile software applications for Android, iOS operating systems.
Чому можна навчитися (результати навчання)	To know: <ul style="list-style-type: none"> <li>– features of mobile software application development;</li> <li>– life cycle of mobile software;</li> <li>– programming basics of Java, Swift languages;</li> <li>– basics of mobile software application design;</li> <li>– basics of mobile software application development;</li> <li>– basics of mobile software application testing;</li> <li>– features of mobile software application exploitation.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Be able to: <ul style="list-style-type: none"> <li>– specify requirements to mobile software applications;</li> <li>– develop logical static structure of mobile software applications;</li> <li>– develop dynamic static structure of mobile software applications;</li> <li>– develop physical structure of mobile software applications;</li> <li>– implement mobile software applications according to their structures.</li> </ul>
Інформаційне забезпечення	Lecture notes Presentations Methodical instructions for laboratory work Methodical instructions for independent work
Форма проведення занять	Lectures, computer classes, independent work of students, consultations
Семестровий контроль	Test

## 9. Розроблення вебзастосунків

Дисципліна	Розроблення вебзастосунків
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	<ul style="list-style-type: none"> <li>– володіння знаннями із дисциплін «Алгоритми та структури даних», «Основи програмування»;</li> <li>– досвід роботи із git.</li> </ul>
<b>Що буде вивчатися</b>	Архітектура вебдодатків, протокол HTTP, основи мови програмування JavaScript, платформа NodeJS, мови HTML і CSS, взаємодія з базами даних, використання та створення HTTP API, вивчення сучасних технологій розроблення вебдодатків. Утиліти node, npm, mongod, Postman.
<b>Чому це цікаво/треба вивчати</b>	<p>Метою дисципліни є оволодіння всім стеком технологій для розроблення повноцінного сучасного вебдодатку: від взаємодії з базою даних та хостингу вебсервера у вебі до динамічної взаємодії з користувачем у веб-браузері.</p> <p>Набуті знання можна використовувати для створення будь-якого вебсайту з базою даних, реєстрацією та аутентифікацією користувачів або веб-сервісу з HTTP API.</p>
<b>Чому можна навчитися (результати навчання)</b>	<p>Знати:</p> <ul style="list-style-type: none"> <li>– основні принципи роботи WWW та мережі Інтернет;</li> <li>– основні елементи HTML та CSS стилі;</li> <li>– публікація статичного веб-сайту у вебі;</li> <li>– клієнт-серверна архітектура;</li> <li>– модель TCP/IP та мережеві сокети;</li> <li>– протокол HTTP, принцип роботи, структура запитів та відповідей;</li> <li>– створення вебсервера з шаблонізацією HTML;</li> <li>– формат даних JSON;</li> <li>– обробка даних HTML-форм на сервері;</li> <li>– взаємодія з NoSQL базою даних MongoDB;</li> <li>– хостинг вебсервера та бази даних на віддалених серверах;</li> <li>– ідентифікація, реєстрація, аутентифікація та авторизація користувачів вебдодатку;</li> <li>– HTTP Cookies та WebStorage;</li> <li>– HTTP API, RPC, SOAP, REST;</li> <li>– HTML DOM та браузерний JavaScript;</li> <li>– технологія AJAX, FetchAPI;</li> <li>– використання браузерних JavaScript бібліотек;</li> <li>– шаблонізація HTML у браузері;</li> <li>– використання Social Auth за допомогою OAuth;</li> <li>– основи MVC вебфреймворків. VueJS;</li> <li>– безпека вебдодатків.</li> </ul>

<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Уміти:</p> <ul style="list-style-type: none"> <li>– створювати і стилізувати сторінки вебсайтів;</li> <li>– публікувати статичні вебсайти у вебі;</li> <li>– створювати клієнт-серверну розподілену систему, компоненти якої взаємодіють по протоколу HTTP;</li> <li>– зберігати та керувати даними за допомогою NoSQL бази даних MongoDB;</li> <li>– хостити вебсервер та базу даних на віддалених серверах;</li> <li>– створювати додатки з реєстрацією користувачів та авторизацією доступу до ресурсів вебсайту чи вебсервера;</li> <li>– використовувати Social Auth, яку надають більшість популярних вебсервісів;</li> <li>– використовувати сторонні HTTP API;</li> <li>– створювати та документувати власні HTTP API;</li> <li>– динамічно змінювати вміст вебсторінок;</li> <li>– створювати динамічні вебсайти за допомогою AJAX та MVC вебфреймворків;</li> <li>– створювати захищені вебдодатки.</li> </ul>
<b>Інформаційне забезпечення</b>	<p>Відео-презентації  Конспект лекцій  Методичні вказівки до виконання лабораторних робіт</p>
<b>Форма проведення занять</b>	<p>Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації</p>
<b>Семестровий контроль</b>	<p>Залік</p>

## 10. Web-application Development

Дисципліна	Web-application Development
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 credit
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer Systems Software Department
<b>Вимоги до початку вивчення</b>	<ul style="list-style-type: none"> <li>– knowledge of the disciplines "Algorithms and data structures", "Fundamentals of programming";</li> <li>– experience with git.</li> </ul>
<b>Що буде вивчатися</b>	<p>Web application architecture, HTTP protocol, basics of JavaScript programming, NodeJS platform, HTML and CSS languages, interaction with databases, HTTP API using and creating, study of modern technologies of web application development.</p> <p>Node, npm, mongod, Postman utilities.</p>
<b>Чому це цікаво/треба вивчати</b>	<p>The aim of the discipline is to master the technology stack for developing full value modern web application: from database interaction and web server hosting in the web to dynamic interaction with the user in web browser.</p> <p>Obtained knowledge can be used to develop any website with database, user registration and authentication or web service with HTTP API.</p>
<b>Чому можна навчитися (результати навчання)</b>	<p>To know:</p> <ul style="list-style-type: none"> <li>– main principles of WWW operation and the Internet network;</li> <li>– main elements of HTML and CSS style;</li> <li>– publication of static website on the web;</li> <li>– client-server architecture;</li> <li>– TCP/IP model and network sockets;</li> <li>– HTTP protocol, operation principle, structure of requests and answers;</li> <li>– creating web server with HTML pattern;</li> <li>– JSON data format;</li> <li>– processing of HTML-form data on server;</li> <li>– interaction with NoSQL for MongoDB database;</li> <li>– hosting of web server and database on remote servers;</li> <li>– identification, registration, authentication and authorization of web application users;</li> <li>– HTTP Cookies and WebStorage;</li> <li>– HTTP API, RPC, SOAP, REST;</li> <li>– HTML DOM and browser JavaScript;</li> <li>– AJAX technology, FetchAPI;</li> <li>– using browser JavaScript libraries;</li> <li>– HTML patterning in browser;</li> <li>– using Social Auth by means of OAuth;</li> <li>– basics of MVC web frameworks. VueJS;</li> <li>– security of web applications.</li> </ul>
<b>Як можна користуватися</b>	<p>Be able to:</p> <ul style="list-style-type: none"> <li>– create and style website pages;</li> </ul>

<b>набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>– publish static websites on the web;</li> <li>– create client-server distributed system, components of which interact by HTTP protocol;</li> <li>– store and manage data using NoSQL for MongoDB database;</li> <li>– host of web server and database on remote servers;</li> <li>– create applications with user registration and access authorization to resources of website or web server;</li> <li>– use Social Auth, which is provided by most popular web services;</li> <li>– use third-party HTTP API;</li> <li>– create and documate own HTTP API;</li> <li>– dynamically change content of web pages;</li> <li>– create dynamic websites using AJAX and MVC web frameworks;</li> <li>– create protected web application.</li> </ul>
<b>Інформаційне забезпечення</b>	<p>Video presentations Lecture notes Methodical instructions for laboratory work</p>
<b>Форма проведення занять</b>	<p>Lectures, computer classes, independent work of students, consultations</p>
<b>Семестровий контроль</b>	<p>Test</p>



## 11. Теорія інформації та кодування

Дисципліна	Теорія інформації та кодування
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями з навчальних дисциплін "Фізика", "Лінійна алгебра та аналітична геометрія", "Комп'ютерна дискретна математика", "Математичний аналіз".
<b>Що буде вивчатися</b>	Основні методи теорії інформації та теорії кодування.
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є набуття студентами теоретичних знань: <ul style="list-style-type: none"> <li>– основ теорії інформації, її основних постулатів та кількісного вимірювання інформації;</li> <li>– теорії кодування інформації в комп'ютерних системах; та практичних навиків:</li> <li>– здатності вибору інформаційної технології відповідно до вимог до інформаційної системи;</li> <li>– виміру кількісних параметрів інформації, обробці, перетворенню, квантуванню та дискретизації сигналів ;</li> <li>– кодування комп'ютерної інформації.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– основні положення, терміни та визначення теорії інформації та кодування;</li> <li>– принципи кодування та декодування дискретної інформації;</li> <li>– методики шифрування та стиснення дискретної інформації;</li> <li>– основних принципи перетворення інформації;</li> <li>– фізичні основи принципів перетворення інформації, перетворення Фур'є;</li> <li>– основи квантової інформації.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– оцінювати кількісні характеристики інформації при обробці, кодуванні, збереженні та передаванні інформації;</li> <li>– використовувати основні способи кодування сигналів з метою реалізації ефективної (безнадлишкової) та завадостійкої передачі даних по мережах та їх обробці в комп'ютерних системах;</li> <li>– вибирати коди, які найкращим чином підходять для конкретних умов їх застосування.</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації

<b>Семестровий контроль</b>	Залік
-----------------------------	-------

## 12. Information Theory and Coding

Дисципліна	Information Theory and Coding
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 credits
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	Knowledge of the disciplines "Physics", "Linear Algebra and Analytical Geometry", "Computer Discrete Mathematics", "Mathematical Analysis".
Що буде вивчатися	Basic methods of information theory and coding theory.
Чому це цікаво/треба вивчати	<p>The purpose of the discipline is to provide students with theoretical knowledge:</p> <ul style="list-style-type: none"> <li>– basics of information theory, its basic postulates and quantitative measurement of information;</li> <li>– information coding theories in computer systems;</li> </ul> <p>and practical skills:</p> <ul style="list-style-type: none"> <li>– ability to choose information technology in accordance with the requirements of the information system;</li> <li>– measurement of quantitative parameters of information, processing, conversion, quantization and sampling of signals;</li> <li>– encoding of computer information.</li> </ul>
Чому можна навчитися (результати навчання)	<p>To know:</p> <ul style="list-style-type: none"> <li>– basic provisions, terms and definitions of information theory and coding;</li> <li>– principles of coding and decoding of discrete information;</li> <li>– methods of encryption and compression of discrete information;</li> <li>– basic principles of information transformation;</li> <li>– physical foundations of the principles of information transformation, Fourier transform;</li> <li>– basics of quantum information.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>To be able:</p> <ul style="list-style-type: none"> <li>– evaluate the quantitative characteristics of information in the processing, encoding, storage and transmission of information;</li> <li>– use the basic methods of signal coding in order to implement efficient (redundant) and noise-tolerant data transmission over networks and their processing in computer systems;</li> <li>– choose codes that are best suited to the specific conditions of their use.</li> </ul>
Інформаційне забезпечення	<p>Lecture notes Presentations Methodical instructions for laboratory work Methodical instructions for independent work</p>
Форма проведення занять	Lectures, computer classes, independent work of students, consultations

<b>Семестровый контроль</b>	Test
-----------------------------	------

### 13. Управління стартап-проєктами програмного забезпечення

<b>Дисципліна</b>	<b>Управління стартап-проєктами програмного забезпечення</b>
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<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 концепцій та принципів створення інноваційних стартапів, формування практичних навичок управління інформаційним продуктом, навичок роботи з споживачами та формування плану розвитку інформаційного продукту.
<b>Чому це цікаво/треба вивчати</b>	Стартап-проєкти є новою організаційно-управлінською формою створення та розвитку програмних проєктів. Вивчаючи цю дисципліну можна набути професійний досвід: <ul style="list-style-type: none"> <li>– планування інформаційного продукту;</li> <li>– створення та просування власного інформаційного продукту;</li> <li>– проведення опитування споживачів та аналіз результатів;</li> <li>– розробка мінімально цінного продукту (прототипу);</li> <li>– проведення презентації власного стартапу інвестору.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	Знання: <ul style="list-style-type: none"> <li>– принципи побудови та моделювання бізнес-моделі стартапу;</li> <li>– методики розробки та просування інформаційного продукту;</li> <li>– методики роботи з споживчими властивостями інформаційного продукту;</li> <li>– методики проведення опитування споживачів та аналіз результатів;</li> <li>– принципи побудови роботи з зацікавленими сторонами;</li> </ul> Уміння: <ul style="list-style-type: none"> <li>– аналізувати проблеми зацікавлених сторін та вимоги до програмного забезпечення;</li> <li>– розробки бізнес-моделі;</li> <li>– проводити маркетингові дослідження інформаційного продукту;</li> <li>– розробити власний мінімально цінний інформаційний продукт (прототип);</li> <li>– підготувати та провести презентацію для зацікавлених сторін.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Засвоєння навчальної дисципліни дозволить набути фахові компетенції: <ul style="list-style-type: none"> <li>– пошук та формування ідеї стартапу;</li> <li>– розробка бізнес-моделі стартапу;</li> <li>– проведення опитувань споживачів та формування споживчих властивостей інформаційного продукту;</li> <li>– розробка мінімально цінного продукту ;</li> <li>– проведення тестування його споживачами;</li> <li>– побудова плану розвитку інформаційного продукту;</li> <li>– підготовка презентації проєкту для інвестора.</li> </ul>

<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт
<b>Форма проведення занять</b>	Формат тренінгу та мастер-класів, лекцій та практичних занять
<b>Семестровий контроль</b>	Залік

#### 14. Management of Software Startup Projects

<b>Дисципліна</b>	<b>Management of Software Startup Projects</b>
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 credits
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer systems software department
<b>Вимоги до початку вивчення</b>	General knowledge of the computer engineering disciplines
<b>Що буде вивчатися</b>	Concepts and principles of innovation startup creation, formation of practical skills to manage information product, formation of practical skills to operate with consumers, formation of information product development plan.
<b>Чому це цікаво/треба вивчати</b>	Startup projects are new organizational and managerial form of creating and developing software projects. Studying this discipline you be able to: <ul style="list-style-type: none"> <li>– plan information product;</li> <li>– develop and promote own information product;</li> <li>– operate consumer surveys and analysis of results;</li> <li>– develop prototype of information product;</li> <li>– conduct presentation of own startup for investor.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– principles of building and modelling business model of startup;</li> <li>– techniques of developing and promoting information product;</li> <li>– techniques of working with consumer properties of information product;</li> <li>– techniques of consumer surveys and result analysis;</li> <li>– principles of building work with stakeholders;</li> </ul> Be able to: <ul style="list-style-type: none"> <li>– analyze problems of stakeholders and requirements to software;</li> <li>– develop business model;</li> <li>– research information product by marketing way;</li> <li>– develop prototype of information product;</li> <li>– prepare and conduct presentation for stakeholders.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Be able to: <ul style="list-style-type: none"> <li>– research and form startup idea;</li> <li>– develop business model of startup;</li> <li>– interview consumers and to form consumer properties of information product;</li> <li>– develop prototype of information product;</li> <li>– operate testing of information product prototype by consumers;</li> <li>– build plan of information product development;</li> <li>– prepare project presentation for investor.</li> </ul>
<b>Інформаційне забезпечення</b>	Lecture notes Presentations Methodical instructions for laboratory work

<b>Форма проведения занятий</b>	Training and master classes, lectures, practical classes
<b>Семестровый контроль</b>	Test



## 15. Технології оброблення великих даних

Дисципліна	Технології оброблення великих даних
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 кредиті ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями із дисциплін «Алгоритми та структури даних», «Основи програмування», «Операційні системи сімейства Linux», курс «Бази даних»
<b>Що буде вивчатися</b>	Загальні принципи та підходи до оброблення великих даних, Hadoop екосистема, розподілені файлові системи, Apache Spark в обробці великих даних та підготовці даних для машинного навчання. Hadoop eco components, Apache Spark, Ambari (для встановлення кластера).
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є вивчення методів та засобів обробки великих послідовностей даних. Планується вивчення таких інструментів як Apache Kafka, Apache Spark, Hadoop + MapReduce та ін.
<b>Чому можна навчитися (результати навчання)</b>	Студенти оволодіють теоретичними та практичними навичками використання різних компонентів екосистеми Hadoop: Hadoop, Yarn, MapReduce, Pig, Hive, Impala, HBase, ZooKeeper, Oozie, Sqoop and Flume та ін.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Набуті навички та уміння можна використовувати для подальшої сертифікації чи працевлаштування у ролі інженера з обробки великих даних
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 16. Big Data Technologies

Дисципліна	Big Data Technologies
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 credits
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer systems software departament
<b>Вимоги до початку вивчення</b>	Strong knowledge of «Algorithms and data structures», «Basics of programming», «Operating systems», «Databases»
<b>Що буде вивчатися</b>	General principles and approaches to big data processing, Hadoop ecosystem, distributed file systems, Apache Spark in big data and in preparing data for machine learning. Hadoop eco components, Apache Spark, Ambari (for cluster installation).
<b>Чому це цікаво/треба вивчати</b>	The purpose of the discipline is to study the methods and means of processing large sequences of data. It is planned to study such tools as Apache Kafka, Apache Spark, Hadoop + MapReduce and others.
<b>Чому можна навчитися (результати навчання)</b>	Students will master the theoretical and practical skills of using various components of the Hadoop ecosystem: Hadoop, Yarn, MapReduce, Pig, Hive, Impala, HBase, ZooKeeper, Oozie, Sqoop and Flume and others.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Acquired skills can be used for further certification or employment as a big data engineer.
<b>Інформаційне забезпечення</b>	Lecture notes Presentations Methodical instructions for laboratory work
<b>Форма проведення занять</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровий контроль</b>	Credit

## 17. Абстрактна алгебра для інженерії програмного забезпечення

Дисципліна	Абстрактна алгебра для інженерії програмного забезпечення
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредитів ЄКТС
Мова викладання	Українська
Кафедра	Програмного забезпечення комп'ютерних систем
Вимоги до початку вивчення	<ul style="list-style-type: none"> <li>– володіння знаннями із дисциплін «Алгоритми та структури даних», «Програмування», «Вища математика»;</li> <li>– досвід розроблення програмного забезпечення на будь-якій мові програмування.</li> </ul>
Що буде вивчатися	Класифікація методів та алгоритмів оптимізації. Обчислювальна складність задач оптимізації. Клас алгоритмів детермінованого локального пошуку. Клас алгоритмів стохастичного локального пошуку. Генетичні алгоритми. Мурашині методи та алгоритми.
Чому це цікаво/треба вивчати	Метою даної дисципліни є формування компетентного спеціаліста в області абстрактної алгебри для інженерії програмного забезпечення, здатного застосовувати і розвивати основні положення і методи дисципліни у виробничий діяльності, самостійно аналізувати будову алгебраїчних об'єктів, будувати математичні моделі, застосовувати апарат дисципліни до вивчення абстрактних алгебраїчних структур. Важливими завданнями є формування в студентів алгебраїчної і теоретико-числової культури, сприяння розвитку логічного та аналітичного мислення студентів, забезпечення інформацією студентів щодо напрямків розвитку сучасної математики, зокрема математики, яка використовується в криптографічних методах захисту інформації.
Чому можна навчитися (результати навчання)	<p>Знати:</p> <ul style="list-style-type: none"> <li>– основні твердження і теореми алгебри та теорії чисел;</li> <li>– основні поняття, зокрема такі як множина, відношення, відображення, комплексне число, ланцюговий дріб, алгебраїчна структура, напівгрупа, моноїд, квазігрупа, група, абелева група, симетрична та знаковмінна групи, порядок елемента групи, циклічна група, періодична група, підгрупа, множина твірних групи, ліві і праві класи розбиття групи за підгрупою, індекс підгрупи, нормальна підгрупа, факторгрупа, кільце, комутативне кільце, кільце з одиницею, дільник нуля, дільник одиниці, найбільший спільний дільник і найменше спільне кратне елементів кільця, квадратичний лишок та нелишок, символ Лежандра, алгебраїчне та трансцендентне число, поле, характеристика поля, розширення поля, поліном, симетричний поліном, поле розкладення полінома.</li> </ul> <p>Уміти:</p> <ul style="list-style-type: none"> <li>– застосовувати стандартні методи і алгоритми алгебри та теорії чисел при розв'язанні задач за допомогою програмних засобів;</li> <li>– наводити приклади, які демонструють сутність теоретичних понять, фактів або спростовують хибні твердження;</li> </ul>

	<ul style="list-style-type: none"> <li>– перевіряти, чи є задана алгебраїчна структура напівгрупою, моноїдом, квазігрупою чи групою;</li> <li>– знаходити порядок елемента групи;</li> <li>– встановлювати ізоморфізм груп;</li> <li>– ділити з остачею елементи евклідового кільця, знаходити дільники елемента кільця, обчислювати найбільший спільний дільник елементів кільця та розробляти програмне забезпечення для виконання цих операцій;</li> <li>– користуватися теоремою Ейлера для знаходження остач від ділення;</li> <li>– користуватися критерієм Ейлера та символом Лежандра при розробленні програмних систем комп'ютерної математики;</li> <li>– розв'язувати системи лінійних конгруенцій з невідомими;</li> <li>– будувати прості розширення полів, знаходити степінь розширення, виконувати арифметичні дії у скінченних розширеннях полів, будувати поле розкладу многочлена.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Розроблення та ефективне застосування програмних додатків для виконання математичних операцій, які використовуються в системах захисту інформації.
<b>Інформаційне забезпечення</b>	Завдання на лабораторні роботи
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 18. Abstract Algebra for Software Engineering

Дисципліна	Abstract Algebra for Software Engineering
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 credit
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	<ul style="list-style-type: none"> <li>– knowledge of the disciplines "Data Structures and Algorithms", "Programming", "Higher Mathematics";</li> <li>– experience in software development for any programming language.</li> </ul>
Що буде вивчатися	Classification of optimization methods and algorithms. Computational complexity of optimization tasks. Algorithms class of deterministic local search. Algorithms class of stochastic local search. Genetic algorithms. Ant methods and algorithms.
Чому це цікаво/треба вивчати	<p>The aim of disciplines is to form competent specialist in field of abstract algebra for software engineering, able to apply and develop basic principles and methods of discipline in production, independently analyze structure of algebraic objects, to build mathematical models, to apply the apparatus of discipline to the study of abstract algebraic structures.</p> <p>Important tasks are formation of algebraic and theoretical-numerical culture in students, promoting the development of logical and analytical thinking of students, providing information to students on the development of modern mathematics, including mathematics used in cryptographic methods of information security.</p>
Чому можна навчитися (результати навчання)	<p>To know:</p> <ul style="list-style-type: none"> <li>– basic statements and theorems of algebra and number theory;</li> <li>– basic concept in particular set, relation, reflection, complex number, continued fraction, algebraic structure, semigroup, monoid, quasigroup, group, abelian group, symmetric and alternating groups, order of group element, cyclic group, periodic group, subgroup, generating set of group, left and right classes of group division by subgroup, subgroup index, normal subgroup, quotient group, ring, commutative ring, ring with unit, zero divisor, unit divisor, greatest common divisor and least common multiple of ring elements, quadratic residue and nonresidue, Legendre symbol, algebraic and transcendental number, field, characteristic field, field extension, polynomial, symmetric polynomial, polynomial splitting field.</li> </ul> <p>Be able to:</p> <ul style="list-style-type: none"> <li>– apply standard methods and algorithms of algebra and number theory in solving problems by means of software;</li> <li>– give examples that demonstrate essence of theoretical concepts, facts or refute false statements;</li> <li>– check whether given algebraic structure is semigroup, monoid, quasigroup or group;</li> <li>– search order of group element;</li> </ul>

	<ul style="list-style-type: none"> <li>– set isomorphism of groups;</li> <li>– divide with remainder of elements of Euclidean ring, to search divisors of ring element, to compute greatest common divisor of ring elements and to develop software for performing these operations;</li> <li>– use Euler's theorem to search remainders from division;</li> <li>– use Euler's criterion and Legendre symbol in development of computer mathematics software systems;</li> <li>– solve systems of linear congruences with unknowns;</li> <li>– construct simple field extensions, to search extension degree, to perform arithmetic operations in finite field extensions, to build schedule field of polynomial.</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Development and effective application of software applications to perform mathematical operations used in information security systems.
<b>Інформаційне забезпечення</b>	Methodical instructions for laboratory work
<b>Форма проведення занять</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровий контроль</b>	Test

## 19. Кросплатформне програмування

Дисципліна	Кросплатформне програмування
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 кредити ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями з навчальних дисциплін «Архітектура комп'ютера», «Операційні системи», «Основи програмування», «Об'єктно-орієнтоване програмування», «Бази даних».
<b>Що буде вивчатися</b>	Кросплатформність. Рівні кросплатформності. Кросплатформні мови програмування (Java, Python). Специфікація вимог до кросплатформних програмного забезпечення. Статична та динамічна логічні структури кросплатформного програмного забезпечення. Фізична структура кросплатформного програмного забезпечення. Програмне реалізування кросплатформних застосунків за їх структурами. Мови програмування Java, Python.
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є набуття теоретичних знань і практичних навиків розроблення кросплатформного програмного забезпечення. Використання набутих практичних навиків дозволить проєктувати, розробляти та експлуатувати кросплатформне програмне забезпечення на прикладі мов програмування Java, Python.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– поняття, рівні кросплатформності;</li> <li>– особливості розроблення кросплатформного програмного забезпечення;</li> <li>– життєвий цикл розроблення кросплатформного програмного забезпечення;</li> <li>– основи програмування мовами Java, Python;</li> <li>– основи проєктування кросплатформного програмного забезпечення;</li> <li>– основи розроблення кросплатформного програмного забезпечення;</li> <li>– основи тестування кросплатформного програмного забезпечення;</li> <li>– особливості експлуатування кросплатформного програмного забезпечення.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– специфікувати вимоги до кросплатформного програмного забезпечення;</li> <li>– створювати статичну логічну структуру кросплатформного програмного забезпечення;</li> <li>– створювати динамічну логічну структуру кросплатформного програмного забезпечення;</li> <li>– створювати фізичну структуру кросплатформного програмного забезпечення;</li> </ul>

	– програмно реалізовувати кросплатформні застосунки.
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт Методичні вказівки до самостійної робіт
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік



## 20. Cross-Platform Programming

Дисципліна	Cross-Platform Programming
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 credits
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer Systems Software Department
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines "Computer Architecture", "Operating systems", "Fundamentals of programming", "Object-oriented programming", "Databases".
<b>Що буде вивчатися</b>	<p>Cross-platform. Levels of cross-platform. Cross-Platform programming languages (Java, Python).</p> <p>Requirement specification for cross-platform software. Static and dynamic logical structures of cross-platform software. Physical structure of cross-platform software. Software implementation of cross-platform applications by their structures.</p> <p>Java and Python programming languages.</p>
<b>Чому це цікаво/треба вивчати</b>	The aim of the discipline is to master the theoretical knowledge and practical skills of developing cross-platform software. Applying the obtained practical skills will allow you to design, develop and operate cross-platform software on the example of Java and Python programming languages.
<b>Чому можна навчитися (результати навчання)</b>	<p>To know:</p> <ul style="list-style-type: none"> <li>– concepts, levels of cross-platform;</li> <li>– features of cross-platform software development;</li> <li>– life cycle of cross-platform software development;</li> <li>– basics of programming by Java, Python languages;</li> <li>– basics of cross-platform software design;</li> <li>– basics of cross-platform software development;</li> <li>– basics of cross-platform software testing;</li> <li>– features of cross-platform software operation.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Be able to:</p> <ul style="list-style-type: none"> <li>– specify cross-platform software requirements;</li> <li>– build static logical structure of cross-platform software;</li> <li>– build dynamic logical structure of cross-platform software;</li> <li>– build physical structure of cross-platform software;</li> <li>– develop cross-platform software applications.</li> </ul>
<b>Інформаційне забезпечення</b>	<p>Lecture notes</p> <p>Presentations</p> <p>Methodical instructions for laboratory work</p> <p>Methodical instructions for independent work</p>

<b>Форма проведения занятий</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровый контроль</b>	Test

## 21. Адміністрування операційних систем Linux

Дисципліна	Адміністрування операційних систем Linux
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС
Мова викладання	Українська
Кафедра	Програмного забезпечення комп'ютерних систем
Вимоги до початку вивчення	Володіння знаннями із дисциплін "Структури та алгоритми даних", "Основи програмування", "Операційні системи". Досвід роботи із ОС Linux на рівні досвідченого користувача (bash, ssh)
Що буде вивчатися	Основи хмарних технологій, AWS (EC2, RDS), bash, MySQL Server, Nginx, Jmeter, SSH.
Чому це цікаво/треба вивчати	Метою дисципліни є отримання навичок автоматизації діяльності із адміністрування ОС, супроводження та тестування вебсервера, та роботи із хмарними сервісами. Набуті знання дозволять користуватися можливостями хмарних сервісів та автоматизувати рутинні завдання в ОС.
Чому можна навчитися (результати навчання)	Знати: <ul style="list-style-type: none"> <li>– принципи адміністрування основних складових ОС;</li> <li>– принципи організації безпечної роботи ОС;</li> <li>– види хмарних сервісів.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Уміти: <ul style="list-style-type: none"> <li>– автоматизувати роботу по адмініструванню та забезпеченню безпеки ОС;</li> <li>– користуватися хмарними сервісами для створення віртуальних машин;</li> <li>– інсталювати вебсервер та розміщувати сайти, тестувати продуктивність вебсерверів.</li> </ul>
Інформаційне забезпечення	Презентації, відеозапис лекцій Методичні вказівки до виконання лабораторних робіт
Форма проведення занять	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
Семестровий контроль	Залік

## 22. Linux Operating Systems Administration

Дисципліна	Linux Operating Systems Administration
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 credit
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	Knowledge of the disciplines "Algorithms and data structures", "Fundamentals of programming". Experience with Linux operating system at the advanced user level (bash, ssh). Knowledge of English.
Що буде вивчатися	Basics of cloud technologies, AWS (EC2, RDS), bash, MySQL server, Nginx, Jmeter, SSH.
Чому це цікаво/треба вивчати	The aim of the discipline is automating OS administration activities, maintaining and testing web servers, and working with cloud services. The acquired knowledge will allow you to use the capabilities of cloud services and automate routine tasks in the OS.
Чому можна навчитися (результати навчання)	<p>Knowledge:</p> <ul style="list-style-type: none"> <li>– principles of administration of the main components of the OS;</li> <li>– principles to secure operating systems;</li> <li>– cloud services types.</li> </ul> <p>To be able to:</p> <ul style="list-style-type: none"> <li>– automate the work on administration and security of the OS;</li> <li>– use cloud services to create and to manage virtual machines;</li> <li>– install a web server and host sites, test the performance of web servers.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Skills:</p> <ul style="list-style-type: none"> <li>– automation the administration tasks and support OS security;</li> <li>– web servers supporting and testing;</li> <li>– cloud services usage.</li> </ul>
Інформаційне забезпечення	Lecture notes, presentations, quizzes Tutorials for laboratory works
Форма проведення занять	Lectures, computer classes, independent work of students, consultations
Семестровий контроль	Test

### 23. Менеджмент програмного продукту

Дисципліна	Менеджмент програмного продукту
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<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 концепцій та принципів менеджменту та маркетингу програмного продукту, формування практичних знань та вмій з розробки стратегії створення та розвитку програмного продукту, здобуття навичок роботи за міжнародними стандартами програмного та проєктного менеджменту.
<b>Чому це цікаво/треба вивчати</b>	Без стратегії створення та виведення програмного продукту на ринок програмних продуктів не можливо досягнуть успіху в проєктах впровадження програмних продуктів. Цей курс дозволить вивчити можливості управління загалом програмним продуктом, а не одної її частиною – розробкою.
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>– методики управління програмним продуктом;</li> <li>– методів логіко-структурного аналізу та методів оцінки ризиків створення програмного продукту;</li> <li>– способів формування стратегії розвитку програмним продуктом;</li> <li>– принципів побудови проєктів в галузі ІТ;</li> <li>– особливостей застосування міжнародних стандартів менеджменту проєктів для створення програмного продукту.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Цей курс рекомендується для слухачів, які планують мати кар'єрний ріст в проєктах ІТ та працювати на рівні менеджерів програмних продуктів, використовуючи суміжні області знань в ІТ індустрії: маркетинг, менеджмент проєктів ІТ. Засвоєння навчальної дисципліни дозволить набути фахові компетенції: <ul style="list-style-type: none"> <li>– створювати стратегію управління новим програмним продуктом ;</li> <li>– формувати та оцінювати ідею інформаційного продукту та можливості впровадження на світових ринках ІТ;</li> <li>– аналізування вимоги до інформаційної програмного продукту та функціональних можливостей з боку зацікавлених сторін проєктів;</li> <li>– обирати інформаційну технологію відповідно до визначених вимог до інформаційної системи;</li> <li>– виконувати повний цикл управління програмним продуктом.</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації

<b>Семестровий контроль</b>	Залік
-----------------------------	-------

## 24. Software Product Management

Дисципліна	Software Product Management
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 credit
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer Systems Software Department
<b>Вимоги до початку вивчення</b>	Possession of general knowledge of the computer engineering disciplines
<b>Що буде вивчатися</b>	Concepts and principles of software product management and marketing, formation of practical knowledge and skills to develop strategies of creating and developing software product. Formatting practical skills according to international standard of programming and project management.
<b>Чому це цікаво/треба вивчати</b>	It isn't possible to succeed in software product implementation projects without strategy of creating and bringing software product to software product market. This course will allow to study possibilities of managing software product in general, rather than one part of it - development.
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– methods of software product management;</li> <li>– methods of logical-structural analysis and methods of risk estimation of software product development;</li> <li>– techniques of forming strategies to develop software product;</li> <li>– principles of designing projects in IT field;</li> <li>– features of applying international standards of project management to develop software product.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	This course is recommended for students which plan to have career in IT projects and to work at level of software product managers, using knowledge of related areas in IT industry: marketing, IT project management. Be able to: <ul style="list-style-type: none"> <li>– develop strategies of new software product management;</li> <li>– form and estimate idea of information product and implementation possibilities in international IT markets;</li> <li>– analyze requirements to information software product and functional possibilities by project stakeholders;</li> <li>– choose information technology according to defined requirement to information system;</li> <li>– perform full cycle of software product management.</li> </ul>
<b>Інформаційне забезпечення</b>	Lecture notes Presentations Methodical instructions for laboratory work
<b>Форма проведення занять</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровий контроль</b>	Test

## 25. Web-технології та розробка web-систем

Дисципліна	Web-технології та розробка web-систем
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредити ЄКТС
Мова викладання	Українська
Кафедра	Програмного забезпечення комп'ютерних систем
Вимоги до початку вивчення	Володіння знаннями із дисциплін "Алгоритми та структури даних", "Програмування", "Об'єктно-орієнтоване програмування". Досвід розроблення програмного забезпечення на будь-якій мові програмування.
Що буде вивчатися	Основи web-розробки засобами HTML, CSS та JavaScript. Семантика та доступність web-сторінок. Оптимізація перформансу web-сторінок та їх безпека. Використання сучасних JavaScript бібліотек та фреймворків для створення web-систем (React).
Чому це цікаво/треба вивчати	Дисципліну підготовлено фахівцями компанії ЕПАМ. Метою дисципліни є формування у студента системи знань та вмінь з використання сучасних підходів та технологій для розробки web-сайтів, розуміння структури та етапів створення web-сторінок, освоєння сучасних засобів комп'ютерних технологій в реалізації web-проектів.
Чому можна навчитися (результати навчання)	Знати: <ul style="list-style-type: none"> <li>– роль HTML, CSS та JavaScript у web-розробці;</li> <li>– основні підходи до розробки web-систем;</li> <li>– способи оптимізації web-сторінок;</li> <li>– сучасні бібліотеки та фреймворки для розробки web-систем.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Уміти: <ul style="list-style-type: none"> <li>– будувати сучасні адаптивні web-сайти на основі технологій (HTML/CSS);</li> <li>– розробляти алгоритми обробки веб-сторінок та динамічного управління HTML-елементами та формами засобами мови JavaScript;</li> <li>– перевіряти та оптимізувати працездатність та безпеку розроблених програм;</li> <li>– створювати сучасні web-системи з використанням бібліотеки React.</li> </ul>
Інформаційне забезпечення	Завдання на лабораторні роботи, відео уроки, презентаційні матеріали
Форма проведення занять	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
Семестровий контроль	Залік



26. Прикладне програмне забезпечення мовою Python

Дисципліна	Прикладне програмне забезпечення мовою Python
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 кредитів ЄКТС
Мова викладання	українська
Кафедра	Програмного забезпечення комп'ютерних систем
Вимоги до початку вивчення	Базові знання з дисциплін «Основи програмування», «Програмування», «Бази даних».
Що буде вивчатися	<p>Курс охоплює всі основи програмування на Python, а також загальні концепції та методи комп'ютерного програмування.</p> <p>Курс знайомить студента з об'єктно-орієнтованим підходом до Python та використанням Python в аналізі даних.</p> <p>Курс розбитий на два модулі.</p> <p>Модуль 1</p> <p>Вступ до програмування на Python. Типи даних, змінні, основні операції введення-виведення. Літерали та оператори Python. Булеві змінні, умовне виконання, цикли, логічні та побітові операції. Прийняття рішень на Python. Списки в розширених додатках.</p> <p>Функції, кортежі, словники та обробка даних у Python. Сфери в Python. Модулі, пакети, рядкові та спискові методи, винятки. Деякі корисні модулі. Пакети. Помилки. Деякі з найбільш корисних винятків.</p> <p>Об'єктно-орієнтований підхід: класи, методи, об'єкти та стандартні ознаки об'єктів; Обробка винятків та робота з файлами. Властивості. Наслідування. Винятки. Генератори та замикання. Обробка файлів. Робота з реальними файлами.</p> <p>Модуль 2</p> <p>Моделі аналітики. Можливості інструменту аналізу даних. Роль Python в аналізі даних. Підготовка даних. Джерела даних. Типи даних та формати.</p> <p>Структури даних Python. Витяг, перетворення та завантаження даних. Поточні та майбутні нормативні акти. Захист даних. Тріада CIA.</p> <p>Форматування даних про час і дату. Директиви щодо форматування дати та часу Python. Читання та запис файлів.</p> <p>Використання основних статистичних методів та методів підготовки даних у Python з Pandas. Взаємодія із зовнішніми застосунками. Аналіз даних за допомогою Python та SQLite.</p>
Чому це цікаво/треба вивчати	<p>Python дуже універсальна, об'єктно-орієнтована мова програмування, яку використовують стартапери та такі технічні гіганти, як Google, Facebook, Dropbox та IBM.</p> <p>Python рекомендується молодим розробникам, які зацікавлені в кар'єрі в галузі безпеки, мереж та Інтернету речей.</p> <p>За допомогою Python можна розробити веб- або настільний додатки, яким могли користуватись мільйонам людей.</p>
Чому можна навчитися	<p>Розв'язання задач за допомогою алгоритмічного підходу в Python.</p> <p>Розуміння роботи програміста в процесі розробки програмного забезпечення за допомогою Python.</p>

<b>(результати навчання)</b>	Взаємодії із зовнішніми застосунками. Аналізувати дані, використовуючи основні статистичні методи та методи підготовки даних у Python з Pandas. Аналізувати дані, використовуючи Python та SQLite.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Набуті навички та уміння можна використовувати для вирішення практичних завдань, пов'язаних з основами програмування на мові Python, а також для розуміння основних понять та прийомів, які використовуються в об'єктно-орієнтованому програмуванні для вирішення основних завдань прикладного аналізу даних.
<b>Інформаційне забезпечення</b>	Методичні вказівки до виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 27. Applied Data Science with Python

Дисципліна	Applied Data Science with Python
Рівень ВО	Перший (бакалаврський)
Курс	3
Обсяг	4 credits
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	Knowledge of the disciplines "Fundamentals of programming", "Programming", "Databases".
Що буде вивчатися	<p>The course covers all the basics of programming in Python, as well as general computer programming concepts and techniques. The course also familiarizes the student with object-oriented approach with Python and using Python in Data Analysis.</p> <p>The course is broken down into two modules.</p> <p>Module 1</p> <p>Introduction to Python Programming. Data Types, Variables, Basic Input-Output Operations. Python literals and operators. Boolean Values, Conditional Execution, Loops, Logical and Bitwise Operations. Making decisions in Python. Lists in advanced applications.</p> <p>Functions, Tuples, Dictionaries, and Data Processing in Python. Scopes in Python.</p> <p>Modules, Packages, String and List Methods, Exceptions. Some useful modules. Packages. Errors. Some of the most useful exceptions.</p> <p>The Object-Oriented Approach: Classes, Methods, Objects, and the Standard Objective Features; Exception Handling, and Working with Files. Properties. Inheritance. Exceptions. Generators and closures. Processing files. Working with real files.</p> <p>Module 2</p> <p>Analytics Models. Data Analytics Tool Capabilities. The Role of Python in Data Analysis. Preparing Data. Sources of Data. Data Types and Formats. Python Data Structures. Extract, Transform, and Load Data. Current and Future Regulations. Data Security. CIA Triad.</p> <p>Formatting Time and Date Data. Python datetime Formatting Directives. Reading and Writing Files.</p> <p>Using basic statistical and data preparation techniques in Python with Pandas. Interacting with External Applications. Internet Meter Data Analysis Working with Python and SQLite.</p>
Чому це цікаво/треба вивчати	<p>How great would it be to design a web or desktop application that millions of people could enjoy? Both are a possibility if you learn how to code in Python. Python is the very versatile, object-oriented programming language used by startups and tech giants, Google, Facebook, Dropbox and IBM. Python is also recommended for aspiring young developers who are interested in pursuing careers in Security, Networking and Internet-of-Things.</p>
Чому можна навчитися	<p>Problem-solve using an algorithmic approach in Python.</p> <p>Understand programmer's work in the software development process using Python.</p>

<b>(результати навчання)</b>	To interact with External Applications. To analyze data using basic statistical and data preparation techniques in Python with Pandas. To analyze data using Python and SQLite.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Students who complete the course will be able to accomplish coding tasks related to the basics of programming in the Python language, and to understand the fundamental notions and techniques used in object-oriented programming to solve the basic tasks of applied data analysis.
<b>Інформаційне забезпечення</b>	Each student has access to hands-on practice materials, labs, quizzes, and assessments to learn how to utilize the skills and knowledge gained on the course and interact with some real-life programming tasks and situations.
<b>Форма проведення занять</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровий контроль</b>	Test

## 28. Аналіз даних

Дисципліна	Аналіз даних
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	<ul style="list-style-type: none"> <li>– володіння знаннями із дисциплін «Алгоритми та структури даних», «Програмування», «Вища математика»;</li> <li>– досвід розроблення програмного забезпечення на будь-якій мові програмування</li> </ul>
<b>Що буде вивчатися</b>	Кластерний аналіз. Байєсівський класифікатор. Однофакторний та двофакторний аналіз. Регресійний аналіз. Часові ряди. Практичний аналіз часових рядів. Багатовимірний аналіз. Проблеми ідентифікації швидких процесів. Елементи теорії оброблення сигналів. Методи регуляризації, фільтрації.
<b>Чому це цікаво/треба вивчати</b>	Розповсюдженню методів аналізу даних сприяє можливість застосовувати до такого аналізу обчислювальну техніку. Замовниками такого аналізу можуть бути великі підприємства, державні установи та представники середнього і малого бізнесу. Тому важливим є знання математичних методів аналізу даних та вміння їх використовувати при розробленні програмного забезпечення.
<b>Чому можна навчитися (результати навчання)</b>	<p>Знати:</p> <ul style="list-style-type: none"> <li>– основні поняття теорії аналізу даних;</li> <li>– основні положення одно- і двофакторного аналізу;</li> <li>– методи кореляційного та регресійного аналізу для виявлення взаємозв'язків між ознаками;</li> <li>– принципи використання кластерного аналізу;</li> <li>– методи вивчення часових рядів;</li> <li>– методи регуляризації та фільтрації.</li> </ul> <p>Уміти:</p> <ul style="list-style-type: none"> <li>– із опису предметної області і задач, що розв'язуються, приймати рішення щодо застосування того або іншого методу оброблення даних;</li> <li>– використовувати можливості обчислювальної техніки при та аналізі даних;</li> <li>– розробляти програмне забезпечення для аналізу великих об'ємів даних.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Розроблення та ефективне застосування програмних додатків для аналізу даних.
<b>Інформаційне забезпечення</b>	Завдання на лабораторні роботи

<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 29. Data Analysis

<b>Дисципліна</b>	<b>Data Analysis</b>
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	3
<b>Обсяг</b>	4 credits
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer Systems Software Department
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines "Algorithms and data structures", "Programming", "Higher mathematics". Software development experience
<b>Що буде вивчатися</b>	Cluster analysis. Bayes classifier. Single-factor analysis of variance and double-factor analysis of variance. Regression analysis. Time series. Practical analysis of time series. Multidimensional analysis. Identification problems of fast processes. Elements of signal processing theory. Regularization and filtration methods.
<b>Чому це цікаво/треба вивчати</b>	Opportunity of applying computer technology to data analysis promote distribution of data analysis methods. Clients of such analysis can be large enterprises, government agencies and representatives of medium and small businesses. Therefore, the knowledge of mathematical data analysis methods and the ability to use them for software developing are important.
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– basic concepts of data analysis theory;</li> <li>– basic provisions of single-factor and double-factor analysis of variance;</li> <li>– methods of correlation and regression analysis to identify relationships between features;</li> <li>– principles of cluster analysis using;</li> <li>– methods of time series studying;</li> <li>– methods of regularization and filtration.</li> </ul> Be able to: <ul style="list-style-type: none"> <li>– make decision on application of feature data processing method depending on description of subject area and solved tasks;</li> <li>– use the capabilities of computer technology in data analysis;</li> <li>– develop the software for large amounts data analysis.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Be able to develop and effective apply of software applications for data analysis.
<b>Інформаційне забезпечення</b>	Methodical instructions for laboratory work
<b>Форма проведення занять</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровий контроль</b>	Test

## Анотації вибірових дисциплін для 4 курсу

### 1. Програмні методи оброблення експериментальних даних

Дисципліна	Програмні методи оброблення експериментальних даних
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 кредитів ЄКТС
Мова викладання	Українська
Кафедра	Програмного забезпечення комп'ютерних систем
Вимоги до початку вивчення	<ul style="list-style-type: none"> <li>– володіння знаннями із дисциплін «Алгоритми та структури даних», «Програмування», «Вища математика»;</li> <li>– досвід розроблення програмного забезпечення на будь-якій мові програмування</li> </ul>
Що буде вивчатися	Метод найменших квадратів. Метод ковзної середньої. Метод експоненційного згладжування. Статистичний аналіз багаторазових вимірювань з випадковими похибками за допомогою програмних засобів. Інтерполяційний многочлен Лагранжа. Скінченно-різницеві інтерполяційні формули. Інтерполяційна формула Ньютона для нерівновіддалених вузлів. Чисельне диференціювання. Чисельне інтегрування. Інтерполяція сплайнами. Програмне забезпечення обробки даних.
Чому це цікаво/треба вивчати	Вивчення даної дисципліни формує у студентів теоретичні та практичні основи статистично-програмної обробки даних експериментальних досліджень та уміння обирати адекватні програмно-математичні методи оброблення експериментальних даних.
Чому можна навчитися (результати навчання)	<p>Знати:</p> <ul style="list-style-type: none"> <li>– базові прогностичні методи;</li> <li>– методи апроксимації даних;</li> <li>– програмні методи інтерполювання даних, що задані таблицею.</li> </ul> <p>Уміти:</p> <ul style="list-style-type: none"> <li>– розробляти програмні системи для реалізації прогностичних моделей;</li> <li>– застосовувати на практиці методи апроксимації даних;</li> <li>– розробляти програмне забезпечення для реалізації методів згладжування даних.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розроблення та ефективного застосування програмних додатків для реалізації прогностичних моделей і програмного забезпечення для апроксимації та згладжування експериментальних даних.
Інформаційне забезпечення	Завдання на лабораторні роботи
Форма проведення занять	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації



<b>Семестровий контроль</b>	Залік
-----------------------------	-------

## 2. Software Methods of Experimental Data Processing

<b>Дисципліна</b>	<b>Software Methods of Experimental Data Processing</b>
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 credits
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer systems software department
<b>Вимоги до початку вивчення</b>	<ul style="list-style-type: none"> <li>– knowledge of the disciplines "Algorithms and data structures", "Programming", "Higher mathematics";</li> <li>– software development experience</li> </ul>
<b>Що буде вивчатися</b>	Least squares method. Moving average method. Exponential smoothing method. Statistical analysis of multiple measurements with random errors by means of software means. Lagrange interpolation polynomials. Finite-difference interpolation formulas. Newton interpolation formula for unequal spacing of nodes. Numerical differentiation. Numerical integration. Spline interpolation. Data processing software.
<b>Чому це цікаво/треба вивчати</b>	Study of this discipline forms theoretical and practical foundations of statistical and software data processing of experimental research, also the ability to choose adequate software and mathematical methods of experimental data processing.
<b>Чому можна навчитися (результати навчання)</b>	<p>To know:</p> <ul style="list-style-type: none"> <li>– basic forecasting models;</li> <li>– methods of data approximating;</li> <li>– software methods of data interpolation given by table.</li> </ul> <p>Be able to:</p> <ul style="list-style-type: none"> <li>– develop software systems to implement forecasting models;</li> <li>– apply methods of data approximating in practice;</li> <li>– develop software to implement methods of data smoothing.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Be able to develop and effective apply of software applications for implementing forecasting models and software for approximating and smoothing experimental data.
<b>Інформаційне забезпечення</b>	Methodical instructions for laboratory work
<b>Форма проведення занять</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровий контроль</b>	Test

### 3. Бізнес-аналіз процесів автоматизації програмного забезпечення

<b>Дисципліна</b>	<b>Бізнес-аналіз процесів автоматизації програмного забезпечення</b>
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<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 концепцій та принципів автоматизації в управлінні інформаційними процесами згідно міжнародним стандартом бізнес-аналізу в проєктах розробки програмного забезпечення. Формування практичних фахових вмінь бізнес-аналітика процесів автоматизації.
<b>Чому це цікаво/треба вивчати</b>	Без розуміння основних концепцій бізнес-аналізу та бізнес-аналітики не може існувати не один фахівець з розробки програмного забезпечення. Цей курс дозволить набути досвід: <ul style="list-style-type: none"> <li>– процесного моделювання та проєктування;</li> <li>– побудови архітектури процесів програмного забезпечення;</li> <li>– бізнес-аналізу інформаційних систем.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	Знання: <ul style="list-style-type: none"> <li>– методів та наукових підходів до моделювання процесів;</li> <li>– методів логіко-структурного аналізу;</li> <li>– способів формування концепції системи процесів;</li> <li>– методів проєктування процесів;</li> <li>– особливостей застосування міжнародних стандартів управлінню бізнес-процесами.</li> </ul> Уміння: <ul style="list-style-type: none"> <li>– визначати вимоги до управління процесами;</li> <li>– моделювати та аналізувати процеси;</li> <li>– проєктувати та оптимізувати процеси;</li> <li>– планування дій до автоматизації бізнес-процесів</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Засвоєння навчальної дисципліни дозволить набути фахові компетенції: <ul style="list-style-type: none"> <li>– комунікації з замовниками програмного забезпечення;</li> <li>– збір вимог до процесів автоматизації;</li> <li>– моделювання та аналіз процесів автоматизації;</li> <li>– проєктування процесів автоматизації;</li> <li>– управління процесами ефективної автоматизації;</li> <li>– обирати інформаційну технологію відповідно до визначених вимог до інформаційної системи;</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт

<b>Форма проведення занять</b>	Формат тренінгу та мастер-класів, лекцій та практичних занять
<b>Семестровий контроль</b>	Залік

#### 4. Business Analysis of Software Automation Processes

<b>Дисципліна</b>	<b>Business Analysis of Software Automation Processes</b>
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 credit
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer Systems Software Department
<b>Вимоги до початку вивчення</b>	Possession of general knowledge of the computer engineering disciplines
<b>Що буде вивчатися</b>	<p>Concepts and principles of automation in information process manager according to international standard of business analysis in software development projects.</p> <p>Formation of practical professional skills of business analytics of automation processes.</p>
<b>Чому це цікаво/треба вивчати</b>	<p>Without understanding of basic concepts of business analysis and business intelligence, there can be more than one software development specialist. This course will provide experience:</p> <ul style="list-style-type: none"> <li>– process modeling and design;</li> <li>– of building architecture of software process;</li> <li>– business analysis of information systems.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	<p>To know:</p> <ul style="list-style-type: none"> <li>– methods and scientific approaches to process modeling;</li> <li>– methods of logical-structural analysis;</li> <li>– techniques to form concepts of process system;</li> <li>– methods of process design;</li> <li>– features of applying international standards for business process management</li> </ul> <p>Be able to:</p> <ul style="list-style-type: none"> <li>– define requirements to process management;</li> <li>– model and to analyzer processes;</li> <li>– design and optimize processes;</li> <li>– plane actions to automation business processes</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Be able to:</p> <ul style="list-style-type: none"> <li>– communicate with customers of software;</li> <li>– collect requirements to automation processes;</li> <li>– model and analyzer automation processes;</li> <li>– design automation processes;</li> <li>– manager effective automation processes;</li> <li>– choose information technology according to defined requirements of information system.</li> </ul>
<b>Інформаційне забезпечення</b>	<p>Lecture notes</p> <p>Presentations</p> <p>Methodical instructions for laboratory work</p>
<b>Форма проведення занять</b>	Training and master classes, lectures, laboratory

<b>Семестровый контроль</b>	Test
-----------------------------	------

## 5. Розроблення програмного забезпечення за методологією Agile

<b>Дисципліна</b>	<b>Розроблення програмного забезпечення за методологією Agile</b>
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 кредити
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями із дисциплін «Алгоритми та структури даних», «Основи програмування», «Якість програмного забезпечення»
<b>Що буде вивчатися</b>	Основні методології розробки програмного забезпечення сімейства Agile. Основні практики Agile та їх застосування Trello tool для сортування та моніторингу виконання робіт
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є оволодіння навичками планування, розподілення, виконання та контролю якості робіт проекту розробки програмного забезпечення згідно методологій Agile. Набуті знання можна використовувати для роботи в проектах більшості ІТ-компаній з розробки програмного забезпечення.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– особливості розробки програмного забезпечення згідно кожної з основних методологій Agile (DSDM, XP, SCRUM, Kanban, Lean, Crystal);</li> <li>– планування за допомогою Planning Pocker;</li> <li>– project backlog, user story;</li> <li>– time-boxing, story boarding, story points;</li> <li>– релізи програмного забезпечення згідно Agile.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– складати список історій користувача та проектний беклог в команді;</li> <li>– оцінювати історії користувача та аналізувати термін її виконання;</li> <li>– планувати командні спринти;</li> <li>– визначати та усувати всі витрати та бар'єри розробки продукту;</li> <li>– планувати початковий реліз та подальші релізи розвитку продукт.</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт Власні дистанційні ресурси Сторінка курсу в Moodle
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 6. Agile Software Development

Дисципліна	Agile Software Development
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 credits
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	Knowledge of the disciplines "Data Structures and Algorithms", "Fundamentals of Programming", "Software Quality"
Що буде вивчатися	Agile family software development basic methodologies. Basic Agile practices and their application Trello tool for work units sorting and monitoring.
Чому це цікаво/треба вивчати	The aim of the discipline is to master the skills of planning, distribution, execution and quality control of software development projects according to Agile methodologies. The acquired knowledges can be used to work on the projects of most software development companies.
Чому можна навчитися (результати навчання)	To know: <ul style="list-style-type: none"> <li>– software development features according to each of the main Agile methodologies (DSDM, XP, SCRUM, Kanban, Lean, Crystal);</li> <li>– Planning Pocker;</li> <li>– project backlog, user story;</li> <li>– time-boxing, story boarding, story points;</li> <li>– Agile software releases.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Be able to: <ul style="list-style-type: none"> <li>– make a list of user stories and project backlog in the team;</li> <li>– estimate user stories and analyze the deadline;</li> <li>– plan team sprints;</li> <li>– identify and eliminate all costs and barriers to product development;</li> <li>– plan the initial release and subsequent product development releases.</li> </ul>
Інформаційне забезпечення	Lecture notes Presentations Methodical instructions for laboratory work
Форма проведення занять	Lectures, computer classes, independent work of students, consultations
Семестровий контроль	Test



## 7. Технології DevOps

Дисципліна	Технології DevOps
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями з навчальних дисциплін «Основи програмування», «Операційні системи».
<b>Що буде вивчатися</b>	Засоби та способи безперервної інтеграції та доставки коду на прикладі популярних CI/CD застосунків (Jenkins, GitlabCI) в хмару AWS з дотриманням стандартів інфраструктури як коду, контейнеризації, конфігурування та оркестрування. Стандарти реліз-менеджменту, розгортання та підтримка декількох середовищ, інтеграція тестування та створення пайпланів.
<b>Чому це цікаво/треба вивчати</b>	Курс базується на практичному досвіді впровадження хмарних DevOps практик та CI/CD пайплайнів для великих корпорацій (клієнтів компанії GlobalLogic) та читається досвідченим Lead DevOps інженером. На сьогоднішній день ці практики є промисловим стандартом та значно прискорюють доставку нових можливостей користувачам. Глибоке розуміння хмарних технологій та досвід створення пайпланів вимагається від всіх інженерів для забезпечення підтримки та доставки коду. Лабораторні роботи побудовані на реальних корпоративних задачах з автоматизації.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– можливості сучасних систем безперервної інтеграції та доставки;</li> <li>– кращі практики з менеджменту доставки коду (Release management);</li> <li>– способи збереження то доставки секретів;</li> <li>– особливості створення пайпланів для різних систем безперервної інтеграції;</li> <li>– способи навантажувального тестування;</li> <li>– засоби розгортання середовищ за вимогою;</li> <li>– способи моніторингу додатків.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– застосовувати практики безперервної інтеграції;</li> <li>– налаштовувати ПЗ для безперервної інтеграції;</li> <li>– створювати пайплайни для доставки коду;</li> <li>– автоматично розгортати середовища для застосунків;</li> <li>– ефективно адмініструвати системи на базі ОС Linux;</li> <li>– застосовувати підхід інфраструктура-як-код;</li> <li>– застосовувати підхід конфігурація-як-код;</li> <li>– ефективно контейнеризувати додатки;</li> <li>– забезпечувати безперервний моніторинг додатків.</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації

	Методичні вказівки до виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 8. DevOps Technologies

Дисципліна	DevOps Technologies
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 credits
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer Systems Software Department
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines "Operating systems", "Fundamentals of programming".
<b>Що буде вивчатися</b>	Means and methods of continuous integration and delivery of code based on popular CICD tools (Jenkins, GitlabCI) to the AWS cloud with compliance with infrastructure-as-code standards, containerization, configuration and orchestration. Release management standards, deployment and support of multiple environments, QA integration and pipelines creation.
<b>Чому це цікаво/треба вивчати</b>	The course is based on practical implementation experience, cloud DevOps practices and CICD pipelines creation for large corporations (GlobalLogic customers) and delivered by experienced Lead DevOps engineer. Today, these practices are the industry standard and significantly accelerate the delivery of new features to users. All IT engineers are required to have a deep understanding of cloud technologies and experience in pipelines creation to support delivery. Laboratory work is based on real corporate customers automation tasks.
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– capabilities of modern systems of continuous integration and delivery;</li> <li>– best practices in code delivery management (Release management);</li> <li>– handling secrets;</li> <li>– features of pipelines for various systems of continuous integration;</li> <li>– methods of load testing;</li> <li>– on-demand environments deployment tools;</li> <li>– applications monitoring.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	To be able: <ul style="list-style-type: none"> <li>– apply CICD practices;</li> <li>– configure CICD Tools;</li> <li>– create code delivery pipelines;</li> <li>– automatically create environments;</li> <li>– effectively administration of Linux-based systems;</li> <li>– apply the infrastructure-as-code approach;</li> <li>– apply a configuration-as-code approach;</li> <li>– effectively containerize applications;</li> <li>– apply continuous monitoring for applications.</li> </ul>
<b>Інформаційне забезпечення</b>	Lecture notes Presentations Methodical instructions for laboratory work Methodical instructions for independent work

<b>Форма проведения занятий</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровый контроль</b>	Test

## 9. Менеджмент проєктів ІТ

Дисципліна	Менеджмент проєктів ІТ
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 кредити ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<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 концепцій та принципів проєктного менеджменту, формування практичних навичок управління проєктами створення програмного забезпечення, здобуття навичок роботи за міжнародними стандартами проєктного менеджменту.
<b>Чому це цікаво/треба вивчати</b>	Вміння управляти проєктами в ІТ є основною компетенцією спеціалістів в цій галузі. Розробка та впровадження програмного забезпечення потребує наступного досвіду: <ul style="list-style-type: none"> <li>– планування проєктів та їх виконання в MS Project;</li> <li>– досвід застосування технологій Agile, Scrum;</li> <li>– розробки плану проєкту;</li> <li>– створення концепції проєкту та її презентація;</li> <li>– створення команди проєктів та управління такою командою;</li> <li>– застосування методології проєктного менеджменту до конкретних проєктів.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	Знання: <ul style="list-style-type: none"> <li>– методики управління проєктами програмного забезпечення;</li> <li>– методів логіко-структурного аналізу та методів оцінки ризиків;</li> <li>– способів формування концепції проєкту;</li> <li>– принципів побудови проєктів в галузі ІТ;</li> <li>– особливостей застосування міжнародних стандартів менеджменту проєктів для створення програмного забезпечення.</li> </ul> Уміння: <ul style="list-style-type: none"> <li>– аналізувати проблеми зацікавлених сторін та вимоги до програмного забезпечення;</li> <li>– оцінювати ризики проєкту та способи подолання цих ризиків;</li> <li>– приймати рішення щодо проєкту в умовах високої невизначеності.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Засвоєння навчальної дисципліни дозволить набути фахові компетенції: <ul style="list-style-type: none"> <li>– аналізувати вимоги до інформаційної системи та функціональних можливостей з боку зацікавлених сторін проєктів;</li> <li>– обирати інформаційну технологію відповідно до визначених вимог до інформаційної системи;</li> <li>– виконувати ініціювання, планування, виконання та завершення проєктів програмного забезпечення за міжнародними стандартами;</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації

	Методичні вказівки до виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 10. IT Project Management

Дисципліна	IT Project Management
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 credits
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	General knowledge of the computer engineering disciplines
Що буде вивчатися	<p>Concepts and principles of project management.</p> <p>Management methods of software development projects.</p> <p>Features of applying international standards for project management to develop software.</p>
Чому це цікаво/треба вивчати	<p>The ability to manage projects in IT is the main competence of specialists in this field. Software development and implement require by the following experience:</p> <ul style="list-style-type: none"> <li>– project planning and its implementation in MS Project;</li> <li>– experience of applying Agile, Scrum technologies;</li> <li>– development of project plan;</li> <li>– development of project concept and its presentation;</li> <li>– creating project team and managing such team;</li> <li>– applying project management methodology for feature projects.</li> </ul>
Чому можна навчитися (результати навчання)	<p>To know:</p> <ul style="list-style-type: none"> <li>– management methods of software development projects;</li> <li>– methods of logical-structural analysis and risk assessment methods;</li> <li>– techniques of project concept forming;</li> <li>– principles of project build in IT field;</li> <li>– features of applying international standards for project management to develop software.</li> </ul> <p>Be able to:</p> <ul style="list-style-type: none"> <li>– analyze stakeholder problems and software requirements;</li> <li>– assess project risks and ways to overcome these risks;</li> <li>– make decisions for project in conditions of high uncertainty.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<p>Be able to:</p> <ul style="list-style-type: none"> <li>– analyze requirements to information system and functional opportunities on the part of project stakeholders;</li> <li>– choose information technology in accordance with certain requirements for information system;</li> <li>– initiate, plan, execute and complete software projects according to international standards;</li> </ul>
Інформаційне забезпечення	<p>Lecture notes</p> <p>Presentations</p> <p>Methodical instructions for laboratory work</p>
Форма проведення занять	Lectures, computer classes, independent work of students, consultations

<b>Семестровый контроль</b>	Test
-----------------------------	------



## 11. Інфраструктура програмного забезпечення

Дисципліна	Інфраструктура програмного забезпечення
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 кредити
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями з навчальних дисциплін «Архітектура комп'ютера», «Організація комп'ютерних мереж», «Операційні системи», «Основи програмування», «Об'єктно-орієнтоване програмування», «Бази даних».
<b>Що буде вивчатися</b>	Інфраструктура як код. Архітектура контейнерного програмного забезпечення. Життєвий цикл розроблення контейнерного програмного забезпечення. Контейнеризування і оркестрування програмного забезпечення (Docker, Kubernetes). Керування конфігуруванням програмного забезпечення (Ansible, Chef, Salt, Puppet). Забезпечення інфраструктури (Terraform, Cloud formation).
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є набуття теоретичних знань і практичних навиків розроблення контейнерного програмного забезпечення. Використання набутих практичних навиків дозволить проектувати, розробляти та експлуатувати контейнерне програмне забезпечення.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– основи програмування інфраструктури (інфраструктура як код);</li> <li>– архітектуру контейнерного програмного забезпечення;</li> <li>– життєвий цикл розроблення контейнерного програмного забезпечення;</li> <li>– основи контейнеризування і оркестрування програмного забезпечення (Docker, Kubernetes);</li> <li>– основи керування конфігуруванням програмного забезпечення (Ansible, Chef, Salt, Puppet);</li> <li>– основи забезпечення інфраструктури (Terraform, Cloud formation).</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– специфікувати вимоги до контейнерного програмного забезпечення;</li> <li>– створювати статичну логічну структуру контейнерного програмного забезпечення;</li> <li>– створювати динамічну логічну структуру контейнерного програмного забезпечення;</li> <li>– створювати фізичну структуру контейнерного програмного забезпечення;</li> <li>– програмно реалізовувати контейнерного програмного забезпечення.</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт Методичні вказівки до самостійної робіт

<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 12. Software Infrastructure

Дисципліна	Software Infrastructure
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 credits
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer Systems Software Department
<b>Вимоги до початку вивчення</b>	Knowledge of the disciplines "Computer Architecture", "Computer Network Organization", "Operating Systems", "Fundamentals of Programming", "Object-oriented programming", "Database".
<b>Що буде вивчатися</b>	Infrastructure as code. Container software architecture. Development life cycle of container software. Containerization and orchestration of software (Docker, Kubernetes). Software configuration management (Ansible, Chef, Salt, Puppet). Infrastructure provision (Terraform, Cloud formation).
<b>Чому це цікаво/треба вивчати</b>	The purpose of discipline is to acquire theoretical knowledge and practical skills of container software development. Using acquired practical skills will allow you to design, develop and operate container software.
<b>Чому можна навчитися (результати навчання)</b>	To know: <ul style="list-style-type: none"> <li>– basics of infrastructure programming (infrastructure as code);</li> <li>– architecture of container software;</li> <li>– life cycle of developing container software;</li> <li>– basics of software containerization and orchestration (Docker, Kubernetes);</li> <li>– basics of software configuration management (Ansible, Chef, Salt, Puppet);</li> <li>– basics of infrastructure provision (Terraform, Cloud formation).</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Be able to: <ul style="list-style-type: none"> <li>– specify requirements to container software;</li> <li>– develop static logic structure of container software;</li> <li>– develop dynamic logic structure of container software;</li> <li>– develop physical structure of container software;</li> <li>– implement of container software.</li> </ul>
<b>Інформаційне забезпечення</b>	Lecture notes Presentations Methodical instructions for laboratory work Methodical instructions for independent work
<b>Форма проведення занять</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровий контроль</b>	Test

### 13. Програмні засоби захисту інформації

Дисципліна	Програмні засоби захисту інформації
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 кредити
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Досвід командного розроблення програмного забезпечення.
<b>Що буде вивчатися</b>	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-Гамалія 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-Гамалія, принципи побудови криптостійких еліптичних кривих, стандарт цифрового підпису (DSS), алгоритм RSA та схема Ель-Гамалія в режимі цифрового підпису, еліптичні алгоритми формування електронного цифрового підпису.
<b>Чому це цікаво/треба вивчати</b>	Після вивчення даної дисципліни у студентів формуються здатності: <ul style="list-style-type: none"> <li>– аналізувати криптографічні системи;</li> <li>– обирати криптографічний алгоритм відповідно до сформульованої задачі;</li> <li>– забезпечувати роботу сучасних криптосистем;</li> <li>– виконувати налагодження та розробку програмного забезпечення для криптографічного захисту.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	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 уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>– проектування систем криптографічного захисту;</li> <li>– розроблення програмних засобів шифрування, дешифрування даних та створення електронного цифрового підпису;</li> <li>– створення програмного забезпечення еліптичної криптографії;</li> <li>– застосування криптографічних стандартів при розробленні програмного забезпечення.</li> </ul>
<b>Інформаційне забезпечення</b>	Завдання для виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

#### 14. Software Instruments of Information Protection

Дисципліна	Software Instruments of Information Protection
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 credits
<b>Мова викладання</b>	English
<b>Кафедра</b>	Computer Systems Software Department
<b>Вимоги до початку вивчення</b>	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>	After studying this discipline students will develop the ability to: <ul style="list-style-type: none"> <li>– analyze cryptographic systems;</li> <li>– select cryptographic algorithm according to formulated task;</li> <li>– provision operation of modern cryptosystems;</li> <li>– debug and develop software for cryptographic protection.</li> </ul>
<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> <li>– estimate the stability of cryptographic algorithms;</li> </ul>

	– summarize the obtained experimental results.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Be able to: – design systems of cryptographic protection; – develop software of data encryption and decryption; – develop electronic digital signature; – develop software of elliptical cryptography; – apply cryptographic standards for software development.
<b>Інформаційне забезпечення</b>	Methodical instructions for laboratory work
<b>Форма проведення занять</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровий контроль</b>	Test

## 15. Технологія Block Chain

Дисципліна	Технологія Block Chain
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 кредити
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями з навчальних дисциплін «Архітектура комп'ютера», «Організація комп'ютерних мереж», «Операційні системи», «Основи програмування», «Об'єктно-орієнтоване програмування», «Бази даних».
<b>Що буде вивчатися</b>	Основи, переваги та недоліки використання технології блокчейн, архітектура програмного забезпечення за технологією блокчейн, способи визначення технології блокчейн, практичне використання технології блокчейн, різновиди практичних застосунків технології блокчейн, практичний застосунок Ethereum технології блокчейн, розроблення програмних застосунків на платформі Ethereum мовою Solidity. Програмна платформа Ethereum, мова програмування Solidity.
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є набуття теоретичних знань і практичних навиків розроблення розподілених програмних застосунків за технологією блокчейн. Використання набутих практичних навиків дозволить проектувати, розробляти та експлуатувати розподілені програмні застосунки.
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– основи використання технології блокчейн;</li> <li>– переваги та недоліки використання технології блокчейн;</li> <li>– архітектуру програмного забезпечення за технологією блокчейн;</li> <li>– способи визначення технології блокчейн;</li> <li>– практичне використання технології блокчейн;</li> <li>– різновиди практичних застосунків технології блокчейн;</li> <li>– практичний застосунок Ethereum технології блокчейн;</li> <li>– розроблення програмних застосунків на платформі Ethereum.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– обирати та використовувати програмні застосунки технології блокчейн;</li> <li>– проектувати розподілені програмні застосунки;</li> <li>– розробляти розподілені програмні застосунки;</li> <li>– експлуатувати розподілені програмні застосунки.</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт Методичні вказівки до самостійної робіт
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації



<b>Семестровий контроль</b>	Залік
-----------------------------	-------

## 16. Block Chain Technology

Дисципліна	Block Chain Technology
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 credits
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	Knowledge of the disciplines "Computer Architecture", "Organization of computer networks", "Operating systems", "Fundamentals of programming", "Object-oriented programming", "Databases", "Software development technologies".
Що буде вивчатися	Fundamentals, advantages and disadvantages of using blockchain technology, software architecture for blockchain technology, ways to define blockchain technology, practical use of blockchain technology, types of practical applications of blockchain technology, practical application of Ethereum blockchain technology, development of software applications on Ethereum in Solidity. Ethereum software platform, Solidity programming language
Чому це цікаво/треба вивчати	The purpose of the discipline is to acquire theoretical knowledge and practical skills in developing distributed software applications using blockchain technology. Using the acquired practical skills will allow you to design, develop and operate distributed software applications.
Чому можна навчитися (результати навчання)	To know: <ul style="list-style-type: none"> <li>– fundamentals of using blockchain technology;</li> <li>– advantages and disadvantages of using blockchain technology;</li> <li>– blockchain software architecture;</li> <li>– ways to determine blockchain technology;</li> <li>– practical use of blockchain technology;</li> <li>– types of practical applications of blockchain technology;</li> <li>– practical application of Ethereum blockchain technology;</li> <li>– development of software applications on the Ethereum platform.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	To be able: <ul style="list-style-type: none"> <li>select and use blockchain software applications;</li> <li>design distributed software applications;</li> <li>develop distributed software applications;</li> <li>exploit distributed software applications.</li> </ul>
Інформаційне забезпечення	Lecture notes Presentations Methodical instructions for laboratory work Methodical instructions for independent work
Форма проведення занять	Lectures, computer classes, independent work of students, consultations
Семестровий контроль	Test

## 17. Емпіричні методи програмної інженерії

Дисципліна	Емпіричні методи програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	<ul style="list-style-type: none"> <li>– володіння знаннями із дисциплін «Структури та алгоритми даних», «Програмування», «Вища математика 1. Диференціальне числення. Лінійна алгебра»;</li> <li>– досвід розроблення програмного забезпечення на будь-якій мові програмування</li> </ul>
<b>Що буде вивчатися</b>	Розроблення та ефективне застосування програмних додатків для генерування псевдовипадкових чисел. Розроблення програмного забезпечення для генерування псевдовипадкових чисел із заданим законом розподілу. Розроблення програмного забезпечення для оцінювання випадковості послідовності чисел, що проводиться за допомогою емпіричних критеріїв.
<b>Чому це цікаво/треба вивчати</b>	Числа, які обираються випадковим чином, знаходять багато корисних застосувань: моделювання, програмування, прийняття рішень, математика та розваги. При цьому від якості генераторів випадкових чисел напряду залежить якість результатів, що отримуються. Цю обставину підкреслює цитата відомого вченого Роберта Кав'ю з Національної лабораторії Оук-Рідж "генерування випадкових чисел занадто важливе, щоб залишити його випадковим".
<b>Чому можна навчитися (результати навчання)</b>	<p>Знати:</p> <ul style="list-style-type: none"> <li>– методи генерування псевдовипадкових чисел з рівномірним законом розподілу на заданому інтервалі;</li> <li>– методи генерування псевдовипадкових чисел з похідними від рівномірного законами розподілу із заданими параметрами;</li> <li>– емпіричні критерії оцінки ступеня випадковості послідовності чисел.</li> </ul> <p>Уміти:</p> <ul style="list-style-type: none"> <li>– застосовувати на практиці методи генерування цілих псевдовипадкових чисел із заданим законом розподілу та заданими параметрами;</li> <li>– застосовувати на практиці методи генерування дійсних псевдовипадкових чисел із заданим законом розподілу та заданими параметрами;</li> <li>– застосовувати на практиці емпіричні критерії оцінки випадковості послідовності чисел;</li> <li>– розробляти програмне забезпечення для генерування послідовності псевдовипадкових чисел із заданим законом розподілу та заданими параметрами;</li> <li>– розробляти програмне забезпечення для оцінки випадковості послідовності чисел.</li> </ul>
<b>Як можна користуватися</b>	– проектування генераторів псевдовипадкових послідовностей чисел з рівномірним розподілом;

<b>набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>– проектування генераторів псевдовипадкових послідовностей чисел з довільним законом розподілу;</li> <li>– розроблення програмного забезпечення для оцінки випадковості послідовності чисел.</li> </ul>
<b>Інформаційне забезпечення</b>	Завдання на лабораторні роботи
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 18. Empirical Software Engineering Methods

Дисципліна	Empirical Software Engineering Methods
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 credit
Мова викладання	English
Кафедра	Computer Systems Software Department
Вимоги до початку вивчення	<ul style="list-style-type: none"> <li>– knowledge of the disciplines "Data Structures and Algorithms", "Programming", "Higher Mathematics 1. Differential calculus. Linear algebra";</li> <li>– experience in software development for any programming language</li> </ul>
Що буде вивчатися	Development and effective application of software applications to generate pseudo random numbers. Software development to generate pseudo random numbers with given distribution law. Software development to estimate number sequence randomness that is performed by means of empirical criteria.
Чому це цікаво/треба вивчати	Numbers chosen by random way find many useful applications: modelling, programming, decision-making, mathematics and game. Quality of obtained results depends on quality of generating pseudo random numbers. This fact is underlined by famous scientist quote of Robert Cavie from Oak Ridge National Laboratory: "generating pseudo random numbers is too important to leave it random".
Чому можна навчитися (результати навчання)	<p>To know:</p> <ul style="list-style-type: none"> <li>– methods of generating pseudo random numbers with uniform distribution law at a given interval;</li> <li>– methods of generating pseudo random numbers with derivatives from uniform distribution laws with given parameters;</li> <li>– empirical criteria of estimating degree of number sequence randomness.</li> </ul> <p>Be able to:</p> <ul style="list-style-type: none"> <li>– apply methods of generating pseudo random integer numbers with given distribution law and given parameters in practice;</li> <li>– apply methods of generating pseudo random real numbers with given distribution law and given parameters in practice;</li> <li>– apply empirical criteria of estimating randomness of number sequence in practice;</li> <li>– develop software for generating sequence of pseudo random numbers with given distribution law and given parameters;</li> <li>– develop software for estimating randomness of number sequence.</li> </ul>
Як можна користуватися набутими знаннями і вміннями (компетентності)	<p>Be able to:</p> <ul style="list-style-type: none"> <li>– design pseudo random number generators with uniform distribution;</li> <li>– design pseudo random number generators with arbitrary distribution law;</li> <li>– develop software to estimate random of number sequence.</li> </ul>
Інформаційне забезпечення	Methodical instructions for laboratory work

<b>Форма проведения занятий</b>	Lectures, computer classes, independent work of students, consultations
<b>Семестровый контроль</b>	Test

## 19. Основи хмарних технологій

Дисципліна	Основи хмарних технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 кредити
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями із дисциплін "Алгоритми та структури даних", "Основи програмування", "Операційні системи". Досвід роботи із ОС Linux та Windows на рівні досвідченого користувача (CLI).
<b>Що буде вивчатися</b>	Хмарні платформи Azure, AWS.
<b>Чому це цікаво/треба вивчати</b>	Метою дисципліни є вивчення основних сервісів популярних хмарних платформ Azure та AWS. Набуті знання дозволять користуватися популярними можливостями хмарних сервісів.
<b>Чому можна навчитися (результати навчання)</b>	Знання: <ul style="list-style-type: none"> <li>– переваг, концепцій, типів хмарних обчислень;</li> <li>– основних служб Azure, AWS;</li> <li>– основних рішень та інструментів керування в Azure, AWS;</li> <li>– принципів управління витратами Azure, AWS.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміння: <ul style="list-style-type: none"> <li>– керувати витратами;</li> <li>– керувати віртуальними машинами;</li> <li>– керувати віртуальними мережами;</li> <li>– керувати сховищами;</li> <li>– проводити моніторинг ресурсів.</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Методичні вказівки до виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 20. Cloud Technologies Fundamentals

Дисципліна	Cloud Technologies Fundamentals
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 credits
Мова викладання	English
Кафедра	Computer Systems Software Departmenet
Вимоги до початку вивчення	Knowledge of the disciplines "Algorithms and data structures", "Fundamentals of programming", "Operating systems". Experience with Windows or Linux Operating systems and ssh. Knowledge of English.
Що буде вивчатися	Azure, AWS cloud platforms
Чому це цікаво/треба вивчати	The aim of the discipline is to study the main services of the popular cloud platforms Azure and AWS. Acquired knowledge will allow you to use the popular capabilities of cloud services.
Чому можна навчитися (результати навчання)	To know: <ul style="list-style-type: none"> <li>– benefits, principles, types of cloud computing;</li> <li>– main Azure, AWS Services;</li> <li>– main solutions and management tools in Azure, AWS;</li> <li>– Azure, AWS cost management principles.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Be able to: <ul style="list-style-type: none"> <li>– manage costs;</li> <li>– manage virtual machines;</li> <li>– manage virtual networks;</li> <li>– manage storages;</li> <li>– monitor resources.</li> </ul>
Інформаційне забезпечення	Lecture notes, presentations, quizzes Tutorials for laboratory works
Форма проведення занять	Lectures, computer classes, independent work of students, consultations
Семестровий контроль	Test



21. Методи побудови програмних засобів для матричних обчислень

<b>Дисципліна</b>	<b>Методи побудови програмних засобів для матричних обчислень</b>
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Курс</b>	4
<b>Обсяг</b>	4 кредити
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем
<b>Вимоги до початку вивчення</b>	Володіння знаннями з навчальних дисциплін «Вища математика 1. Диференціальне числення. Лінійна алгебра», «Алгоритми та структури даних». Досвід командного розроблення програмного забезпечення.
<b>Що буде вивчатися</b>	Сучасні методи та програмні засоби для роботи з матрицями великої розмірності та їх місце в математичному моделюванні.
<b>Чому це цікаво/треба вивчати</b>	Після вивчення даної дисципліни у студентів формуються здатності: <ul style="list-style-type: none"> <li>– аналізувати програмні методи виконання матричних обчислень;</li> <li>– аналізувати та оцінювати похибки проведених обчислень над матрицями великої розмірності;</li> <li>– виконувати вибір необхідних математичних та програмних функцій для розв'язання поставленої задачі;</li> <li>– розробляти та ефективно застосувати програмні додатки для чисельного розв'язання матричних задач.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	Знати: <ul style="list-style-type: none"> <li>– методи розв'язання часткової проблеми власних значень;</li> <li>– методи розв'язання повної проблеми власних значень;</li> <li>– метод виконання сингулярного розкладення прямокутної матриці;</li> <li>– методи апроксимації таблично-заданих функцій;</li> <li>– методи програмного розв'язання систем нелінійних рівнянь.</li> </ul> Уміти: <ul style="list-style-type: none"> <li>– аналізувати та оцінювати похибку програмно отриманого розв'язку математичної задачі;</li> <li>– узагальнювати та аналізувати програмні методи розв'язання систем нелінійних рівнянь;</li> <li>– застосовувати на практиці методи розв'язання математичних задач на ЕОМ;</li> <li>– розробляти програмне забезпечення для чисельного розв'язання матричних задач.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Уміти: <ul style="list-style-type: none"> <li>– проєктувати системи комп'ютерної математики для виконання матричних обчислень;</li> <li>– використовувати сучасні математичні пакети.</li> </ul>
<b>Інформаційне забезпечення</b>	Конспект лекцій Презентації Завдання для виконання лабораторних робіт

<b>Форма проведення занять</b>	Лекційні заняття, комп'ютерний практикум, самостійна робота студентів, консультації
<b>Семестровий контроль</b>	Залік

## 22. Matrix Calculus Software Methods

Дисципліна	Matrix Calculus Software Methods
Рівень ВО	Перший (бакалаврський)
Курс	4
Обсяг	4 credits
Мова викладання	English
Кафедра	Computer systems software department
Вимоги до початку вивчення	Knowledge of the disciplines "Higher Mathematics 1. Differential calculus. Linear Algebra ", "Algorithms and Data Structures". Experience in team software development.
Що буде вивчатися	Modern methods and software for working with big dimensional matrices and their place in mathematical modeling.
Чому це цікаво/треба вивчати	After studying this discipline, students be able to: <ul style="list-style-type: none"> <li>– to analyze software methods of performing matrix calculations;</li> <li>– to analyze and estimate the errors of the calculations on the matrices of big dimensions;</li> <li>– to perform the selection of the necessary mathematical and software functions to solve the problem;</li> <li>– to develop and effectively apply software applications for numerical solution of matrix problems.</li> </ul>
Чому можна навчитися (результати навчання)	To know: <ul style="list-style-type: none"> <li>– methods of solving a partial problem of eigenvalues;</li> <li>– methods of solving the complete problem of eigenvalues;</li> <li>– method of performing singular decomposition of a rectangular matrix;</li> <li>– methods of approximation of tabular - given functions;</li> <li>– software methods for solving of systems of nonlinear equations.</li> </ul> Be able to: <ul style="list-style-type: none"> <li>– analyze and estimate the error of the software obtained solution of the mathematical problem;</li> <li>– generalize and analyze software methods for solving systems of nonlinear equations;</li> <li>– apply in practice methods of solving mathematical problems on a computer;</li> <li>– develop software for numerical solution of matrix problems.</li> </ul>
Як можна користуватися набутими знаннями і умінями (компетентності)	Be able to: <ul style="list-style-type: none"> <li>– design of computer mathematics systems for performing matrix calculations;</li> <li>– use of modern mathematical packages.</li> </ul>
Інформаційне забезпечення	Lecture notes Presentations Methodical instructions for laboratory work
Форма проведення занять	Lectures, computer classes, independent work of students, consultations

<b>Семестровый контроль</b>	Test
-----------------------------	------