



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

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

ЗАТВЕРДЖЕНО

Методичною радою
КПІ ім. Ігоря Сікорського
(протокол № 5 від «05» 03. 2026р.)

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 2023, 2024 року)

УХВАЛЕНО:

Вченою радою ФПСІМ
КПІ ім. Ігоря Сікорського
(протокол № 8 від «26» січня 2026 р.)

Київ – 2026

Розробники Ф-каталогу

Шкурат Оксана Сергіївна, к.т.н., доцент кафедри програмного забезпечення комп'ютерних систем, факультет програмних систем та прикладної математики (ФПСІМ)

Ф-каталог розглянуто та погоджено на засіданні кафедри програмного забезпечення комп'ютерних систем, протокол № 9 від 21.01.2026 р.

Зміст

| | |
|---|----|
| Інструкція користувачам каталогу | 5 |
| Ф-Каталог | 6 |
| Анотації вибірових дисциплін для 3 курсу | |
| 1. Обчислювальні методи оптимізації | 8 |
| 2. Computational Optimization Methods | 9 |
| 3. Розроблення мобільних застосунків | 10 |
| 4. Mobile Application Development | 11 |
| 5. Розроблення вебзастосунків | 12 |
| 6. Web Application Development | 14 |
| 7. Теорія інформації та кодування | 26 |
| 8. Information Theory and Coding | 27 |
| 9. Управління стартап проектами програмного забезпечення | 18 |
| 10. Management of Software Startup Projects | 19 |
| 11. Технології оброблення великих даних | 20 |
| 12. Big Data Technologies | 21 |
| 13. Абстрактна алгебра для інженерії програмного забезпечення | 22 |
| 14. Abstract Algebra for Software Engineering | 24 |
| 15. Функційне програмування | 26 |
| 16. Functional Programming | 28 |
| 17. Кросплатформне програмування | 29 |
| 18. Cross-Platform Programming | 31 |
| 19. Системне програмування | 32 |
| 20. System Programming | 33 |
| 21. Менеджмент програмного продукту | 34 |
| 22. Software Product Management | 35 |
| 23. Web-технології та розробка web-систем | 36 |
| 24. Web Technologies and Web Systems Development | 37 |
| 25. Прикладне програмне забезпечення мовою Python | 38 |
| 26. Python Applied Software Development | 40 |
| 27. Аналіз даних | 42 |
| 28. Data Analysis | 43 |
| Анотації вибірових дисциплін для 4 курсу | |
| 1. Програмні методи оброблення експериментальних даних | 44 |
| 2. Software Methods of Experimental Data Processing | 45 |
| 3. Менеджмент проєктів ІТ | 46 |
| 4. IT Project Management | 47 |
| 5. Розроблення програмного забезпечення за методологією Agile | 48 |
| 6. Agile Software Development | 49 |
| 7. Основи хмарних технологій | 50 |
| 8. Cloud Technologies Fundamentals | 51 |
| 9. Обробка та моделювання даних мовою Python | 52 |
| 10. Data Processing and Modeling in Python | 53 |
| 11. Бізнес-аналіз процесів автоматизації програмного забезпечення | 54 |
| 12. Business Analysis of Software Automation Processes | 55 |
| 13. Інфраструктура програмного забезпечення | 56 |
| 14. Software Infrastructure | 57 |
| 15. Програмні засоби захисту інформації | 58 |
| 16. Software Instruments of Information Protection | 60 |
| 17. Технологія Block Chain | 62 |
| 18. Block Chain Technology | 63 |

| | |
|--|----|
| 19. Емпіричні методи програмної інженерії | 64 |
| 20. Empirical Software Engineering Methods | 66 |
| 21. Методи побудови програмних засобів для матричних обчислень | 67 |
| 22. Matrix Calculus Software Methods | 68 |

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 – 24 кредити. У навчальному плані зазначається також семестр, у якому викладається вибіркова дисципліна, форма семестрового контролю, види та обсяги навчальних занять.

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

2.1 Реєстрація студента в my.kpi.ua

2.2 Здійснення студентом вибору дисциплін

2.3 Підтвердження студенту його вибору навчальних дисциплін із Ф-Каталогу

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

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

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

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

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

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

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

Ф-Каталог

Дисципліни для 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. | Обчислювальні методи оптимізації | 5 | 4 | залік |
| 2. | Computational Optimization Methods | 5 | 4 | залік |
| 3. | Розроблення мобільних застосунків | 5 | 4 | залік |
| 4. | Mobile Application Development | 5 | 4 | залік |
| 5. | Розроблення вебзастосунків | 5 | 4 | залік |
| 6. | Web Application Development | 5 | 4 | залік |
| 7. | Теорія інформації та кодування | 5 | 4 | залік |
| 8. | Information Theory and Coding | 5 | 4 | залік |
| 9. | Управління стартап проектами програмного забезпечення | 5 | 4 | залік |
| 10. | Management of Software Startup Projects | 5 | 4 | залік |
| 11. | Технології оброблення великих даних | 5 | 4 | залік |
| 12. | Big Data Technologies | 5 | 4 | залік |
| 13. | Абстрактна алгебра для інженерії програмного забезпечення | 6 | 4 | залік |
| 14. | Abstract Algebra for Software Engineering | 6 | 4 | залік |
| 15. | Функційне програмування | 6 | 4 | залік |
| 16. | Functional Programming | 6 | 4 | залік |
| 17. | Кросплатформне програмування | 6 | 4 | залік |
| 18. | Cross-Platform Programming | 6 | 4 | залік |
| 19. | Системне програмування | 6 | 4 | залік |
| 20. | System Programming | 6 | 4 | залік |
| 21. | Менеджмент програмного продукту | 6 | 4 | залік |
| 22. | Software Product Management | 6 | 4 | залік |
| 23. | Web-технології та розробка web-систем | 6 | 4 | залік |
| 24. | Web Technologies and Web Systems Development | 6 | 4 | залік |
| 25. | Прикладне програмне забезпечення мовою Python | 6 | 4 | залік |
| 26. | Python Applied Software Development | 6 | 4 | залік |
| 27. | Аналіз даних | 6 | 4 | залік |
| 28. | Data Analysis | 6 | 4 | залік |

Дисципліни для 4 курсу (вибір третьокурсниками)

Потрібно обрати 24 кредити за рік: 7-й семестр: 12 кредитів; 8-й семестр: 12 кредитів.

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

Анотації вибіркових дисциплін для 3 курсу

1. Обчислювальні методи оптимізації

| Дисципліна | Обчислювальні методи оптимізації |
|---|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 5 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями з дисциплін «Алгоритми та структури даних», «Програмування», «Математичний аналіз». Досвід розроблення програмного забезпечення на будь-якій мові програмування. |
| 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 | Знати: <ul style="list-style-type: none"> – методи багатокритеріальної оптимізації; – еволютивні методи оптимізації; – стохастичні алгоритми знаходження екстремального значення функції. Уміти: <ul style="list-style-type: none"> – застосовувати на практиці методи багатокритеріальної оптимізації для побудови програмних систем; – застосовувати на практиці мурашині алгоритми для розроблення програмного забезпечення розв'язання математичних задач; – розробляти програмне забезпечення для реалізації стохастичних алгоритмів обчислення екстремального значення функції. |
| Як можна користуватися набутими знаннями і вміннями | Розроблення та ефективного застосування програмних додатків для обчислення екстремальних значень функцій однієї змінної та функції кількох змінних. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

2. Computational Optimization Methods

| Course | Computational Optimization Methods |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 5 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Data Structures and Algorithms", "Programming", "Higher Mathematics". Experience in software development for any programming language. |
| What will be studied | 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. |
| Why is this interesting / worth exploring | Rapid expansion of application scope of mathematical optimization methods forms requirement for solving new classes of increased complexity and dimension tasks in recent years. 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. Obtainment of required training for applying acquired knowledge to study other disciplines. |
| What can you learn | To know: <ul style="list-style-type: none"> – methods of multicriteria optimization; – evolutionary optimization methods; – stochastic algorithms for searching extreme value of function. Be able to: <ul style="list-style-type: none"> – apply methods of multicriteria optimization to develop program systems in practice; – apply ant algorithms to develop software for solving mathematical tasks in practice; – develop software for implementing stochastic algorithms of computing extreme value of function. |
| How to use the acquired knowledge and skills | Development and effective application of software applications to compute extreme values of single-variable functions and several-variables functions. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

3. Розроблення мобільних застосунків

| Дисципліна | Розроблення мобільних застосунків |
|---|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 5 семестр |
| Обсяг дисципліни | 4 кредити ЕКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями з навчальних дисциплін «Основи програмування», «Програмування», «Бази даних». |
| Що буде вивчатися | 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. |
| Чому це цікаво/треба вивчати | 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. |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – особливості розроблення мобільних програмних застосунків; – життєвий цикл мобільного програмного забезпечення; – основи програмування мовами Java, Swift; – основи проектування мобільних програмних застосунків; – основи розроблення мобільних програмних застосунків; – основи тестування мобільних програмних застосунків; – особливості експлуатування мобільних програмних застосунків. |
| Як можна користуватися набутими знаннями і уміннями | Уміти: <ul style="list-style-type: none"> – специфікувати вимоги до мобільних програмних застосунків; – створювати статичну логічну структуру мобільних програмних застосунків; – створювати динамічну логічну структуру мобільних програмних застосунків; – створювати фізичну структуру мобільних програмних застосунків; – програмно реалізовувати мобільні програмні застосунки за їх структурами. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

4. Mobile Application Development

| Course | Mobile Application Development |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 5 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Fundamentals of Programming", "Programming", "Databases". |
| What will be studied | 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. |
| Why is this interesting / worth exploring | 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. |
| What can you learn | To know: <ul style="list-style-type: none"> – features of mobile software application development; – life cycle of mobile software; – programming basics of Java, Swift languages; – basics of mobile software application design; – basics of mobile software application development; – basics of mobile software application testing; – features of mobile software application exploitation. |
| How to use the acquired knowledge and skills | Be able to: <ul style="list-style-type: none"> – specify requirements to mobile software applications; – develop logical static structure of mobile software applications; – develop dynamic static structure of mobile software applications; – develop physical structure of mobile software applications; – implement mobile software applications according to their structures. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

5. Розроблення вебзастосунків

| Дисципліна | Розроблення вебзастосунків |
|-------------------------------------|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 5 семестр |
| Обсяг дисципліни | 4 кредити ЕКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями з дисциплін «Алгоритми та структури даних», «Основи програмування», «Програмування». |
| Що буде вивчатися | Основні концепції вебпрограмування: клієнт-серверна архітектура вебзастосунків, протокол HTTP (/1.1, /2, /3), REST API та специфікація OpenAPI, асинхронне програмування. Розроблення серверної частини вебзастосунків на прикладі платформи .NET та фреймворку ASP.NET Core. Основи розроблення клієнтської частини вебзастосунків: HTML, JavaScript/TypeScript, CSS. |
| Чому це цікаво/треба вивчати | <p>Сучасний світ неможливо уявити без вебзастосунків – від простих інформаційних сайтів до складних інтерактивних сервісів, що використовуються в бізнесі, освіті, медицині та інших сферах. Вивчення розроблення вебзастосунків дає можливість не лише зрозуміти принципи створення сучасних вебрішень, а й опанувати затребувані навички програмування, роботи з базами даних, серверними та клієнтськими технологіями. Опанування цих навичок відкриває широкі можливості для майбутньої кар'єри, адже попит на веброзробників залишається стабільно високим.</p> <p>У ході виконання робіт комп'ютерних практикумів буде створено повноцінний вебзастосунок, який складатиметься із серверного REST API з базою даних та клієнтської частини з використанням сучасних браузерних технологій. При цьому студенти не обмежуються певною мовою програмування чи технологією.</p> <p>Ця дисципліна також буде цікавою студентам, що обрали платформу .NET в якості своєї основної платформи для професійної діяльності, адже викладання відбувається з її використанням у якості прикладу (вебфреймворк ASP.NET Core). Це дозволить поглибити загальні знання платформи .NET (наприклад, асинхронного програмування), які можливо використовувати і при розробленні інших типів застосунків.</p> |
| Чому можна навчитися | <p>Знати:</p> <ul style="list-style-type: none"> – основи протоколу HTTP; – архітектурний стиль REST та специфікації OpenAPI; – принципи асинхронного програмування; – принципи розроблення серверних частин вебзастосунків, від базових основ до API observability та автентифікації і авторизації; – архітектурні шаблони проектування Dependency Injection, ORM, DTO, Object Mapper; – принципи відображення реляційної моделі баз даних у об'єктно-орієнтовану парадигму; – принципи розроблення клієнтських частин вебзастосунків, від шарів розмітки (HTML) та відображення (CSS) до шару поведінки (JavaScript і TypeScript). |

| | |
|--|---|
| Як можна користуватися набутими знаннями і уміннями | Уміти: <ul style="list-style-type: none"> – проектувати серверне API вебзастосунків за сучасними стандартами та вимогами; – розроблювати серверні частини вебзастосунків із використанням сучасних технологій; – розроблювати клієнтські частини вебзастосунків за сучасними браузерними стандартами. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

6. Web Application Development

| Course | Web Application Development |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 5 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Algorithms and Data Structures", "Fundamentals of Programming", "Programming". |
| What will be studied | Basic concepts of web programming: client-server architecture of web applications, HTTP protocol (/1.1, /2, /3), REST API and OpenAPI specification, asynchronous programming. Development of the server-side of web applications using the .NET platform and ASP.NET Core framework as an example. Fundamentals of developing the client-side of web applications: HTML, JavaScript/TypeScript, CSS. |
| Why is this interesting / worth exploring | <p>The modern world is impossible to imagine without web applications – from simple information sites to complex interactive services used in business, education, medicine and other areas. Studying web application development allows you not only to understand the principles of creating modern web solutions, but also to master the required skills of programming, working with databases, server and client technologies. Mastering these skills opens up wide opportunities for a future career, because the demand for web developers remains consistently high.</p> <p>During the computer workshops, a full-fledged web application will be created, which will consist of a server REST API with a database and a client part using modern browser technologies. At the same time, students are not limited to a specific programming language or technology.</p> <p>This discipline will also be interesting for students who have chosen the .NET platform as their main platform for professional activity, as the teaching is done using it as an example (the ASP.NET Core web framework). This will allow you to deepen your general knowledge of the .NET platform (e.g., asynchronous programming), which can be used when developing other types of applications.</p> |
| What can you learn | <p>To know:</p> <ul style="list-style-type: none"> – basics of the HTTP protocol; – REST architectural style and OpenAPI specification; – principles of the asynchronous programming; – principles of developing server-side web applications, from basic fundamentals to API observability and authentication and authorization; – architectural design patterns Dependency Injection, ORM, DTO, Object Mapper; – principles of mapping the relational database model into an object-oriented paradigm; – principles of developing client-side web applications, from the markup (HTML) and presentation (CSS) layers to the behavior layer (JavaScript and TypeScript). |

| | |
|---|---|
| How to use the acquired knowledge and skills | Be able to: <ul style="list-style-type: none"> – design the server API of web applications according to modern standards and requirements; – develop the server-side of web applications using modern technologies; – develop the client-side of web applications according to modern browser standards. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

7. Теорія інформації та кодування

| Дисципліна | Теорія інформації та кодування |
|--|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 5 семестр |
| Обсяг дисципліни | 4 кредити ЕКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями з навчальних дисциплін "Лінійна алгебра та аналітична геометрія", "Комп'ютерна дискретна математика", "Математичний аналіз". |
| Що буде вивчатися | Основні методи теорії інформації та теорії кодування. |
| Чому це цікаво/треба вивчати | Метою дисципліни є набуття студентами теоретичних знань: <ul style="list-style-type: none"> – основ теорії інформації, її основних постулатів та кількісного вимірювання інформації; – теорії кодування інформації в комп'ютерних системах; та практичних навиків: <ul style="list-style-type: none"> – здатності вибору інформаційної технології відповідно до вимог до інформаційної системи; – виміру кількісних параметрів інформації, обробці, перетворенню, квантуванню та дискретизації сигналів ; – кодування комп'ютерної інформації. |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – основні положення, терміни та визначення теорії інформації та кодування; – принципи кодування та декодування дискретної інформації; – методики шифрування та стиснення дискретної інформації; – основних принципи перетворення інформації; – фізичні основи принципів перетворення інформації, перетворення Фур'є; – основи квантової інформації. |
| Як можна користуватися набутими знаннями і уміннями | Уміти: <ul style="list-style-type: none"> – оцінювати кількісні характеристики інформації при обробці, кодуванні, збереженні та передаванні інформації; – використовувати основні способи кодування сигналів з метою реалізації ефективною (безнадлишковою) та завадостійкою передачі даних по мережах та їх обробці в комп'ютерних системах; – вибирати коди, які найкращим чином підходять для конкретних умов їх застосування. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

8. Information Theory and Coding

| Course | Information Theory and Coding |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 5 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Linear Algebra and Analytical Geometry", "Computer Discrete Mathematics", "Mathematical Analysis". |
| What will be studied | Basic methods of information theory and coding theory. |
| Why is this interesting / worth exploring | <p>The purpose of the discipline is to provide students with theoretical knowledge:</p> <ul style="list-style-type: none"> – basics of information theory, its basic postulates and quantitative measurement of information; – information coding theories in computer systems; <p>and practical skills:</p> <ul style="list-style-type: none"> – ability to choose information technology in accordance with the requirements of the information system; – measurement of quantitative parameters of information, processing, conversion, quantization and sampling of signals; – encoding of computer information. |
| What can you learn | <p>To know:</p> <ul style="list-style-type: none"> – basic provisions, terms and definitions of information theory and coding; – principles of coding and decoding of discrete information; – methods of encryption and compression of discrete information; – basic principles of information transformation; – physical foundations of the principles of information transformation, Fourier transform; – basics of quantum information. |
| How to use the acquired knowledge and skills | <p>To be able:</p> <ul style="list-style-type: none"> – evaluate the quantitative characteristics of information in the processing, encoding, storage and transmission of information; – 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; – choose codes that are best suited to the specific conditions of their use. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

9. Управління стартап проектами програмного забезпечення

| Дисципліна | Управління стартап проектами програмного забезпечення |
|--|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 5 семестр |
| Обсяг дисципліни | 4 кредити ЕКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння загальними знаннями із дисциплін комп'ютерної інженерії. |
| Що буде вивчатися | 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. |
| Чому це цікаво/треба вивчати | Стартап проекти є новою організаційно-управлінською формою створення та розвитку програмних проєктів. Вивчаючи цю дисципліну можна набути професійний досвід: планування інформаційного продукту; створення та просування власного інформаційного продукту; проведення опитування споживачів та аналіз результатів; розробка мінімально цінного продукту (прототипу); проведення презентації власного стартапу інвестору. |
| Чому можна навчитися | Знання: <ul style="list-style-type: none"> – принципи побудови та моделювання бізнес-моделі стартапу; – методики розробки та просування інформаційного продукту; – методики роботи з споживчими властивостями інформаційного продукту; – методики проведення опитування споживачів та аналіз результатів; – принципи побудови роботи з зацікавленими сторонами; Уміння: <ul style="list-style-type: none"> – аналізувати проблеми зацікавлених сторін та вимоги до програмного забезпечення; – розробки бізнес-моделі; – проводити маркетингові дослідження інформаційного продукту; – розробити власний мінімально цінний інформаційний продукт (прототип); – підготувати та провести презентацію для зацікавлених сторін. |
| Як можна користуватися набутими знаннями і уміннями | Засвоєння навчальної дисципліни дозволить набути фахові компетенції: <ul style="list-style-type: none"> – пошук та формування ідеї стартапу; – розробка бізнес-моделі стартапу; – проведення опитувань споживачів та формування споживчих властивостей інформаційного продукту; – розробка мінімально цінного продукту ; – проведення тестування його споживачами; – побудова плану розвитку інформаційного продукту; – підготовка презентації проєкту для інвестора. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

10. Management of Software Startup Projects

| Course | Management of Software Startup Projects |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 5 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | General knowledge of the computer engineering disciplines |
| What will be studied | 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. |
| Why is this interesting / worth exploring | Startup projects are new organizational and managerial form of creating and developing software projects. Studying this discipline you be able to: plan information product; develop and promote own information product; operate consumer surveys and analysis of results; develop prototype of information product; conduct presentation of own startup for investor. |
| What can you learn | <p>To know:</p> <ul style="list-style-type: none"> – principles of building and modelling business model of startup; – techniques of developing and promoting information product; – techniques of working with consumer properties of information product; – techniques of consumer surveys and result analysis; – principles of building work with stakeholders; <p>Be able to:</p> <ul style="list-style-type: none"> – analyze problems of stakeholders and requirements to software; – develop business model; – research information product by marketing way; – develop prototype of information product; – prepare and conduct presentation for stakeholders. |
| How to use the acquired knowledge and skills | <p>Be able to:</p> <ul style="list-style-type: none"> – research and form startup idea; – develop business model of startup; – interview consumers and to form consumer properties of information product; – develop prototype of information product; – operate testing of information product prototype by consumers; – build plan of information product development; – prepare project presentation for investor. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

11. Технології оброблення великих даних

| Дисципліна | Технології оброблення великих даних |
|--|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 5 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями із дисциплін «Алгоритми та структури даних», «Основи програмування», «Бази даних». |
| Що буде вивчатися | 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). |
| Чому це цікаво/треба вивчати | Метою дисципліни є вивчення методів та засобів обробки великих послідовностей даних. Планується вивчення таких інструментів як Apache Kafka, Apache Spark, Hadoop + MapReduce та ін. |
| Чому можна навчитися | 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. |
| Як можна користуватися набутими знаннями і уміннями | Набуті навички та уміння можна використовувати для подальшої сертифікації чи працевлаштування у ролі інженера з обробки великих даних. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

12. Big Data Technologies

| Course | Big Data Technologies |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 5 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Strong knowledge of «Algorithms and Data Structures», «Fundamentals of Programming», «Databases» |
| What will be studied | 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). |
| Why is this interesting / worth exploring | 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. |
| What can you learn | 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. |
| How to use the acquired knowledge and skills | Acquired skills can be used for further certification or employment as a big data engineer. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

13. Абстрактна алгебра для інженерії програмного забезпечення

| Дисципліна | Абстрактна алгебра для інженерії програмного забезпечення |
|------------------------------|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 6 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями із дисциплін «Алгоритми та структури даних», «Програмування», «Математичний аналіз». Досвід розроблення програмного забезпечення на будь-якій мові програмування. |
| 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 | Знати: <ul style="list-style-type: none"> – основні твердження і теореми алгебри та теорії чисел; – 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. Уміти: <ul style="list-style-type: none"> – застосовувати стандартні методи і алгоритми алгебри та теорії чисел при розв'язанні задач за допомогою програмних засобів; – наводити приклади, які демонструють сутність теоретичних понять, фактів або спростовують хибні твердження; – перевіряти, чи є задана алгебраїчна структура напівгрупою, моноїдом, квазігрупою чи групою; |

| | |
|--|--|
| | <ul style="list-style-type: none"> – знаходити порядок елемента групи; – встановлювати ізоморфізм груп; – ділити з остачею елементи евклідового кільця, знаходити дільники елемента кільця, обчислювати найбільший спільний дільник елементів кільця та розробляти програмне забезпечення для виконання цих операцій; – користуватися теоремою Ейлера для знаходження остач від ділення; – користуватися критерієм Ейлера та символом Лежандра при розробленні програмних систем комп'ютерної математики; – розв'язувати системи лінійних конгруенцій з невідомими; – будувати прості розширення полів, знаходити степінь розширення, виконувати арифметичні дії у скінченних розширеннях полів, будувати поле розкладу многочлена. |
| Як можна користуватися набутими знаннями і вміннями | Розроблення та ефективне застосування програмних додатків для виконання математичних операцій, які використовуються в системах захисту інформації. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

14. Abstract Algebra for Software Engineering

| Course | Abstract Algebra for Software Engineering |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 6 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Data Structures and Algorithms", "Programming", "Higher Mathematics". Experience in software development for any programming language. |
| What will be studied | 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. |
| Why is this interesting / worth exploring | 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. 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. |
| What can you learn | To know: <ul style="list-style-type: none"> – basic statements and theorems of algebra and number theory; – 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. Be able to: <ul style="list-style-type: none"> – apply standard methods and algorithms of algebra and number theory in solving problems by means of software; – give examples that demonstrate essence of theoretical concepts, facts or refute false statements; – check whether given algebraic structure is semigroup, monoid, quasigroup or group; – search order of group element; |

| | |
|---|--|
| | <ul style="list-style-type: none"> – set isomorphism of groups; – 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; – use Euler's theorem to search remainders from division; – use Euler's criterion and Legendre symbol in development of computer mathematics software systems; – solve systems of linear congruences with unknowns; – construct simple field extensions, to search extension degree, to perform arithmetic operations in finite field extensions, to build schedule field of polynomial. |
| How to use the acquired knowledge and skills | Development and effective application of software applications to perform mathematical operations used in information security systems. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

15. Функційне програмування

| Дисципліна | Функційне програмування |
|--|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 6 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями з дисциплін «Алгоритми та структури даних», «Основи програмування», «Програмування». |
| Що буде вивчатися | Теоретична база парадигми функційного програмування. Основні концепції функційного програмування. Переваги та недоліки функційного програмування. Мова програмування F#. Методи забезпечення якості застосунків, розроблених функційними мовами програмування. Розроблення вебзастосунків функційними мовами програмування. |
| Чому це цікаво/треба вивчати | <p>Парадигма функційного програмування з кожним роком стає все більш популярною, адже певні класи задач значно легше вирішуються саме в цій парадигмі, а не у об'єктно-орієнтованій чи процедурній. Елементи функційного програмування (такі як функції першого класу чи зіставлення зі зразком) на сьогодні додано до багатьох мультипарадигмових мов програмування (C#, Java, JavaScript, тощо). Відповідно зростає і попит на фахівців, що здатні розробляти програмне забезпечення зі застосуванням функційної парадигми, наприклад, в Україні також з'явилися компанії, що використовують функційний набір технологій в якості основного. Тому ґрунтовні знання функційної парадигми програмування можуть бути значною перевагою на ринку праці.</p> <p>Ця дисципліна також буде цікавою студентам, що обрали платформу .NET в якості своєї основної платформи для професійної діяльності, адже викладання відбувається з використанням мови програмування F#, що є частиною платформи .NET.</p> <p>Викладання функційної парадигми відбувається у «прагматичному» стилі, з фокусом саме на практичному застосуванні.</p> |
| Чому можна навчитися | <p>Знання:</p> <ul style="list-style-type: none"> – основних концепцій в основі парадигми функційного програмування; – меж застосовності парадигми функційного програмування на практиці; – основних аспектів розроблення програмного забезпечення функційною мовою програмування F#. |
| Як можна користуватися набутими знаннями і вміннями | <p>Уміння:</p> <ul style="list-style-type: none"> – розроблювати програмне забезпечення різних типів зі застосуванням функційних мов програмування; – супроводжувати програмне забезпечення різних типів, що розроблене із застосуванням функційних мов програмування; – забезпечувати якість розробленого функційними мовами програмного забезпечення; |

| | |
|----------------------------------|--|
| | – розроблювати програмне забезпечення різних типів зі застосуванням елементів функційної парадигми, що доступні в інших мовах програмування. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

16. Functional Programming

| Course | Functional Programming |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 6 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Algorithms and data structures", "Fundamentals of programming", "Programming". |
| What will be studied | The theoretical base of the functional programming paradigm. Basic concepts of functional programming. Advantages and disadvantages of functional programming. F# programming language. Methods of ensuring the quality of applications developed in functional programming languages. Development of web applications in functional programming languages. |
| Why is this interesting / worth exploring | <p>The functional programming paradigm is becoming more and more popular every year, because certain classes of problems are much easier to solve in this paradigm than in object-oriented or procedural ones. Elements of functional programming (such as first-class functions or pattern matching) have been added to many multiparadigm programming languages today (C#, Java, JavaScript, etc.).</p> <p>Accordingly, the demand for specialists capable of developing software using the functional paradigm is also growing, for example, companies that use a functional set of technologies as the main ones have also appeared in Ukraine. Therefore, thorough knowledge of the functional paradigm of programming can be a significant advantage in the labor market.</p> <p>This discipline will also be interesting for students who have chosen the .NET platform as their main platform for professional activity, because the teaching takes place using the F# programming language, which is part of the .NET platform.</p> <p>The functional paradigm is taught in a "pragmatic" style, with a focus on practical application.</p> |
| What can you learn | <p>Knowledge of the:</p> <ul style="list-style-type: none"> – basic concepts of the functional programming paradigm; – limits of applicability of the functional programming paradigm in practice; – main aspects of software development in the functional programming language F#. |
| How to use the acquired knowledge and skills | <p>Skills:</p> <ul style="list-style-type: none"> – develop various types of software using functional programming languages; – support various types of software developed using functional programming languages; – ensure the quality of software developed in functional languages; – develop software of various types using elements of the functional paradigm available in other programming languages. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

17. Кросплатформне програмування

| Дисципліна | Кросплатформне програмування |
|--|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 6 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практик: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями з навчальних дисциплін «Основи програмування», «Програмування», «Бази даних». |
| Що буде вивчатися | 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. |
| Чому це цікаво/треба вивчати | 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. |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – поняття, рівні кросплатформності; – особливості розроблення кросплатформного програмного забезпечення; – життєвий цикл розроблення кросплатформного програмного забезпечення; – основи програмування мовами Java, Python; – основи проєктування кросплатформного програмного забезпечення; – основи розроблення кросплатформного програмного забезпечення; – основи тестування кросплатформного програмного забезпечення; – особливості експлуатування кросплатформного програмного забезпечення. |
| Як можна користуватися набутими знаннями і вміннями | Уміти: <ul style="list-style-type: none"> – специфікувати вимоги до кросплатформного програмного забезпечення; – створювати статичну логічну структуру кросплатформного програмного забезпечення; – створювати динамічну логічну структуру кросплатформного програмного забезпечення; – створювати фізичну структуру кросплатформного програмного забезпечення; – програмно реалізовувати кросплатформні застосунки. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |

| | |
|-----------------------------|-------|
| Семестровий контроль | Залік |
|-----------------------------|-------|

18. Cross-Platform Programming

| Course | Cross-Platform Programming |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 6 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Fundamentals of Programming", "Programming", "Databases". |
| What will be studied | <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> |
| Why is this interesting / worth exploring | 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. |
| What can you learn | <p>To know:</p> <ul style="list-style-type: none"> – concepts, levels of cross-platform; – features of cross-platform software development; – life cycle of cross-platform software development; – basics of programming by Java, Python languages; – basics of cross-platform software design; – basics of cross-platform software development; – basics of cross-platform software testing; – features of cross-platform software operation. |
| How to use the acquired knowledge and skills | <p>Be able to:</p> <ul style="list-style-type: none"> – specify cross-platform software requirements; – build static logical structure of cross-platform software; – build dynamic logical structure of cross-platform software; – build physical structure of cross-platform software; – develop cross-platform software applications. |
| Information support of the course | <p>Syllabus</p> <p>Electronic lecture notes</p> |
| Semester assessment | Final Test |

19. Системне програмування

| Дисципліна | Системне програмування |
|--|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 6 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями з дисциплін «Структури та алгоритми даних», «Основи програмування». Досвід програмування мовою Сі (компілятор gcc). |
| 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 освітнього компонента 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, проєктувати та розробляти системне програмне забезпечення відповідно до поставленого завдання. |
| Чому можна навчитися | Знання: <ul style="list-style-type: none"> – системного підходу до проєктування програмного забезпечення автономних та вбудованих обчислювальних платформ; – сучасних методів проєктування програмного забезпечення обчислювальних платформ у складі електронних пристроїв різного призначення; – ресурсів апаратних платформ та методів керування ними; – програмних інтерфейсів операційних систем. |
| Як можна користуватися набутими знаннями і вміннями | Уміння: <ul style="list-style-type: none"> – користуватись програмними інтерфейсами операційних систем; – аналізувати розподіл ресурсів апаратної платформи та планувати їх використання; – розробляти сервісне програмне забезпечення та елементи операційних систем. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

20. System Programming

| Course | System Programming |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 6 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Algorithms and data structures", "Fundamentals of programming". Experience with C programming (gcc compiler). |
| What will be studied | 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. |
| Why is this interesting / worth exploring | 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. |
| What can you learn | Knowledge of the: <ul style="list-style-type: none"> – system approach to software design of autonomous and embedded computing platforms; – modern methods of designing software for computing platforms as part of electronic devices for various purposes; – hardware platform resources and management methods; – operating system software interfaces. |
| How to use the acquired knowledge and skills | Skills: <ul style="list-style-type: none"> – use software interfaces of operating systems; – analyze the distribution of hardware platform resources and plan their use; – develop service software and operating system elements. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

21. Менеджмент програмного продукту

| Дисципліна | Менеджмент програмного продукту |
|--|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 6 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння загальними знаннями із дисциплін комп'ютерної інженерії |
| Що буде вивчатися | Метою викладання дисципліни є засвоєння студентами концепцій та принципів менеджменту та маркетингу програмного продукту, формування практичних знань та вмінь з розробки стратегії створення та розвитку програмного продукту, здобуття навичок роботи за міжнародними стандартами програмного та проектного менеджменту. |
| Чому це цікаво/треба вивчати | Без стратегії створення та виведення програмного продукту на ринок програмних продуктів не можливо досягнуть успіху в проектах впровадження програмних продуктів. Цей курс рекомендується для слухачів, які планують мати кар'єрний ріст в проектах ІТ та працювати на рівні менеджерів програмних продуктів, використовуючи суміжні області знань в ІТ індустрії: маркетинг, менеджмент проектів ІТ. |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – методики управління програмним продуктом; – методи логіко-структурного аналізу та методи оцінки ризиків створення програмного продукту; – способи формування стратегії розвитку програмним продуктом; – принципи побудови проектів в галузі ІТ; – особливості застосування міжнародних стандартів менеджменту проектів для створення програмного продукту. |
| Як можна користуватися набутими знаннями і уміннями | Засвоєння навчальної дисципліни дозволить набути фахові компетенції: <ul style="list-style-type: none"> – створювати стратегію управління новим програмним продуктом ; – формувати та оцінювати ідею інформаційного продукту та можливості впровадження на світових ринках ІТ; – аналізувати вимоги до інформаційної програмного продукту та функціональних можливостей з боку зацікавлених сторін проектів; – обирати інформаційну технологію відповідно до визначених вимог до інформаційної системи; – виконувати повний цикл управління програмним продуктом. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

22. Software Product Management

| Course | Software Product Management |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 6 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Possession of general knowledge of the computer engineering disciplines |
| What will be studied | 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. |
| Why is this interesting / worth exploring | 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 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. |
| What can you learn | To know: <ul style="list-style-type: none"> – methods of software product management; – methods of logical-structural analysis and methods of risk estimation of software product development; – techniques of forming strategies to develop software product; – principles of designing projects in IT field; – features of applying international standards of project management to develop software product. |
| How to use the acquired knowledge and skills | Be able to: <ul style="list-style-type: none"> – develop strategies of new software product management; – form and estimate idea of information product and implementation possibilities in international IT markets; – analyze requirements to information software product and functional possibilities by project stakeholders; – choose information technology according to defined requirement to information system; – perform full cycle of software product management. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

23. Web-технології та розробка web-систем

| Дисципліна | Web-технології та розробка web-систем |
|--|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 6 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практик: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями із дисциплін "Алгоритми та структури даних", "Основи програмування", "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 | Метою дисципліни є формування у студента системи знань та вмій 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 | Знати: <ul style="list-style-type: none"> – роль HTML, CSS та JavaScript у web-розробці; – основні підходи до розробки web-систем; – способи оптимізації web-сторінок; – сучасні бібліотеки та фреймворки для розробки web-систем. |
| Як можна користуватися набутими знаннями і уміннями | Уміти: <ul style="list-style-type: none"> – будувати сучасні адаптивні web-сайти на основі технологій (HTML/CSS); – розробляти алгоритми обробки веб-сторінок та динамічного управління HTML-елементами та формами засобами мови JavaScript; – перевіряти та оптимізувати працездатність та безпеку розроблених програм; – створювати сучасні web-системи з використанням бібліотеки React. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

24. Web Technologies and Web Systems Development

| Course | Web Technologies and Web Systems Development |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 6 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Algorithms and Data Structures", "Fundamentals of Programming", "Programming". Experience in software development in any programming language. |
| What will be studied | Fundamentals of web development using HTML, CSS and JavaScript. Semantics and accessibility of web pages. Optimizing the performance of web pages and their security. Use of modern JavaScript libraries and frameworks for creating web systems (React). |
| Why is this interesting / worth exploring | The purpose of the discipline is to form the student's system of knowledge and skills in the use of modern approaches and technologies for the development of websites, understanding the structure and stages of creating web pages, mastering modern means of computer technologies in the implementation of web projects. |
| What can you learn | To know: <ul style="list-style-type: none"> – the role of HTML, CSS and JavaScript in web development; – basic approaches to the development of web systems; – approaches to optimize web pages; – modern libraries and frameworks for the development of web systems. |
| How to use the acquired knowledge and skills | Be able to: <ul style="list-style-type: none"> – build modern adaptive websites based on technologies (HTML/CSS); – develop algorithms for processing web pages and dynamic management of HTML elements and forms using the JavaScript language; – check and optimize the performance and security of the developed programs; – create modern web systems using the React library. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

25. Прикладне програмне забезпечення мовою Python

| Дисципліна | Прикладне програмне забезпечення мовою Python |
|------------------------------|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 6 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| 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 дві частини.</p> <p>Частина 1. 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. 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. 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>Частина 2. Моделі аналітики. Можливості інструменту аналізу даних. Роль Python в аналізі даних. Підготовка даних. Джерела даних. Типи даних та формати. Структури даних Python. Витяг, перетворення та завантаження даних. Поточні та майбутні нормативні акти. Захист даних. Тріада CIA. Форматування даних про час і дату. Директиви щодо форматування дати та часу Python. Читання та запис файлів. 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> <p>Взаємодії із зовнішніми застосунками. Аналізувати дані, використовуючи основні статистичні методи та методи підготовки даних у Python з Pandas. Аналізувати дані, використовуючи Python та SQLite.</p> |

| | |
|--|---|
| Як можна користуватися набутими знаннями і уміннями | 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. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

26. Python Applied Software Development

| Course | Python Applied Software Development |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 6 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Fundamentals of Programming", "Programming", "Databases". |
| What will be studied | <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 parts.</p> <p>Part 1. 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. Functions, Tuples, Dictionaries, and Data Processing in Python. Scopes in Python. Modules, Packages, String and List Methods, Exceptions. Some useful modules. Packages. Errors. Some of the most useful exceptions. 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>Part 2. 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. Formatting Time and Date Data. Python datetime Formatting Directives. Reading and Writing Files. 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> |
| Why is this interesting / worth exploring | 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. |
| What can you learn | <p>Problem-solve using an algorithmic approach in Python.</p> <p>Understand programmer's work in the software development process using Python.</p> <p>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.</p> |

| | |
|---|---|
| How to use the acquired knowledge and skills | 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. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

27. Аналіз даних

| Дисципліна | Аналіз даних |
|---|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 3 курс, 6 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями із дисциплін «Алгоритми та структури даних», «Програмування», «Математичний аналіз». Досвід розроблення програмного забезпечення на будь-якій мові програмування |
| Що буде вивчатися | 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. |
| Чому це цікаво/треба вивчати | 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. |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – основні поняття теорії аналізу даних; – основні положення одно- і двофакторного аналізу; – методи кореляційного та регресійного аналізу для виявлення взаємозв'язків між ознаками; – принципи використання кластерного аналізу; – методи вивчення часових рядів; – методи регуляризації та фільтрації. Уміти: <ul style="list-style-type: none"> – із опису предметної області і задач, що розв'язуються, приймати рішення щодо застосування того або іншого методу оброблення даних; – використовувати можливості обчислювальної техніки при та аналізі даних; – розробляти програмне забезпечення для аналізу великих об'ємів даних. |
| Як можна користуватися набутими знаннями і уміннями | Розроблення та ефективне застосування програмних додатків для аналізу даних. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

28. Data Analysis

| Course | Data Analysis |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 3 rd year, 6 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Algorithms and Data Structures", "Programming", "Higher mathematics". Software development experience. |
| What will be studied | 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. |
| Why is this interesting / worth exploring | 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. |
| What can you learn | <p>To know:</p> <ul style="list-style-type: none"> – basic concepts of data analysis theory; – basic provisions of single-factor and double-factor analysis of variance; – methods of correlation and regression analysis to identify relationships between features; – principles of cluster analysis using; – methods of time series studying; – methods of regularization and filtration. <p>Be able to:</p> <ul style="list-style-type: none"> – make decision on application of feature data processing method depending on description of subject area and solved tasks; – use the capabilities of computer technology in data analysis; – develop the software for large amounts data analysis. |
| How to use the acquired knowledge and skills | Be able to develop and effective apply of software applications for data analysis. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

Анотації вибірових дисциплін для 4 курсу

1. Програмні методи оброблення експериментальних даних

| Дисципліна | Програмні методи оброблення експериментальних даних |
|--|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 7 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями із дисциплін «Алгоритми та структури даних», «Програмування», «Математичний аналіз». Досвід розроблення програмного забезпечення на будь-якій мові програмування |
| 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 | Знати: <ul style="list-style-type: none"> – базові прогностичні методи; – методи апроксимації даних; – програмні методи інтерполювання даних, що задані таблицею. Уміти: <ul style="list-style-type: none"> – розробляти програмні системи для реалізації прогностичних моделей; – застосовувати на практиці методи апроксимації даних; – розробляти програмне забезпечення для реалізації методів згладжування даних. |
| Як можна користуватися набутими знаннями і уміннями | Розроблення та ефективне застосування програмних додатків для реалізації прогностичних моделей і програмного забезпечення для апроксимації та згладжування експериментальних даних. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

2. Software Methods of Experimental Data Processing

| Course | Software Methods of Experimental Data Processing |
|--|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 7 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Algorithms and Data Structures", "Programming", "Higher mathematics". Software development experience |
| What will be studied | 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. |
| Why is this interesting / worth exploring | 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. |
| What can you learn | To know: <ul style="list-style-type: none"> – basic forecasting models; – methods of data approximating; – software methods of data interpolation given by table. Be able to: <ul style="list-style-type: none"> – develop software systems to implement forecasting models; – apply methods of data approximating in practice; – develop software to implement methods of data smoothing. |
| How to use the acquired knowledge and skills | Be able to develop and effective apply of software applications for implementing forecasting models and software for approximating and smoothing experimental data. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

3. Менеджмент проєктів ІТ

| Дисципліна | Менеджмент проєктів ІТ |
|--|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 7 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння загальними знаннями із дисциплін комп'ютерної інженерії |
| Що буде вивчатися | 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. |
| Чому це цікаво/треба вивчати | 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"> – планування проєктів та їх виконання в MS Project; – досвід застосування технологій Agile, Scrum; – розробки плану проєкту; – створення концепції проєкту та її презентація; – створення команди проєктів та управління такою командою; – застосування методології проєктного менеджменту до конкретних проєктів. |
| Чому можна навчитися | Знання: <ul style="list-style-type: none"> – методики управління проєктами програмного забезпечення; – методів логіко-структурного аналізу та методів оцінки ризиків; – способів формування концепції проєкту; – принципів побудови проєктів в галузі ІТ; – особливостей застосування міжнародних стандартів менеджменту проєктів для створення програмного забезпечення. Уміння: <ul style="list-style-type: none"> – аналізувати проблеми зацікавлених сторін та вимоги до програмного забезпечення; – оцінювати ризики проєкту та способи подолання цих ризиків; – приймати рішення щодо проєкту в умовах високої невизначеності. |
| Як можна користуватися набутими знаннями і уміннями | Засвоєння навчальної дисципліни дозволить набути фахові компетенції: <ul style="list-style-type: none"> – аналізувати вимоги до інформаційної системи та функціональних можливостей з боку зацікавлених сторін проєктів; – обирати інформаційну технологію відповідно до визначених вимог до інформаційної системи; – виконувати ініціювання, планування, виконання та завершення проєктів програмного забезпечення за міжнародними стандартами; |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

4. IT Project Management

| Course | IT Project Management |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 7 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | General knowledge of the computer engineering disciplines |
| What will be studied | <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> |
| Why is this interesting / worth exploring | <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"> – project planning and its implementation in MS Project; – experience of applying Agile, Scrum technologies; – development of project plan; – development of project concept and its presentation; – creating project team and managing such team; – applying project management methodology for feature projects. |
| What can you learn | <p>To know:</p> <ul style="list-style-type: none"> – management methods of software development projects; – methods of logical-structural analysis and risk assessment methods; – techniques of project concept forming; – principles of project build in IT field; – features of applying international standards for project management to develop software. <p>Be able to:</p> <ul style="list-style-type: none"> – analyze stakeholder problems and software requirements; – assess project risks and ways to overcome these risks; – make decisions for project in conditions of high uncertainty. |
| How to use the acquired knowledge and skills | <p>Be able to:</p> <ul style="list-style-type: none"> – analyze requirements to information system and functional opportunities on the part of project stakeholders; – choose information technology in accordance with certain requirements for information system; – initiate, plan, execute and complete software projects according to international standards; |
| Information support of the course | <p>Syllabus</p> <p>Electronic lecture notes</p> |
| Semester assessment | Final Test |

5. Розроблення програмного забезпечення за методологією Agile

| Дисципліна | Розроблення програмного забезпечення за методологією Agile |
|---|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 7 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практик: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями із дисциплін «Алгоритми та структури даних», «Основи програмування», «Програмування», «Компоненти програмної інженерії». |
| Що буде вивчатися | Основні методології розробки програмного забезпечення сімейства Agile. Основні практики Agile та їх застосування. Trello tool для сортування та моніторингу виконання робіт |
| Чому це цікаво/треба вивчати | 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 проєктах більшості ІТ-компаній з розробки програмного забезпечення. |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – особливості розробки програмного забезпечення згідно кожної з основних методологій Agile (DSDM, XP, SCRUM, Kanban, Lean, Crystal); – планування за допомогою Planning Pocker; – project backlog, user story; – time-boxing, story boarding, story points; – релізи програмного забезпечення згідно Agile. |
| Як можна користуватися набутими знаннями і уміннями | Уміти: <ul style="list-style-type: none"> – складати список історій користувача та проєктний беклог в команді; – оцінювати історії користувача та аналізувати термін її виконання; – планувати командні спринти; – визначати та усувати всі витрати та бар'єри розробки продукту; – планувати початковий реліз та подальші релізи розвитку продукт. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій Сторінка курсу в Moodle |
| Семестровий контроль | Залік |

6. Agile Software Development

| Course | Agile Software Development |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 7 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines " Algorithms and Data Structures", "Fundamentals of Programming", "Programming", "Components of Software Engineering". |
| What will be studied | Agile family software development basic methodologies. Basic Agile practices and their application. Trello tool for work units sorting and monitoring. |
| Why is this interesting / worth exploring | 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. |
| What can you learn | To know: <ul style="list-style-type: none"> – software development features according to each of the main Agile methodologies (DSDM, XP, SCRUM, Kanban, Lean, Crystal); – Planning Pocker; – project backlog, user story; – time-boxing, story boarding, story points; – Agile software releases. |
| How to use the acquired knowledge and skills | Be able to: <ul style="list-style-type: none"> – make a list of user stories and project backlog in the team; – estimate user stories and analyze the deadline; – plan team sprints; – identify and eliminate all costs and barriers to product development; – plan the initial release and subsequent product development releases. |
| Information support of the course | Syllabus Electronic lecture notes Course page in Moodle |
| Semester assessment | Final Test |

7. Основи хмарних технологій

| Дисципліна | Основи хмарних технологій |
|--|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 7 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практик: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями із дисциплін "Основи комп'ютерних систем і мереж". |
| Що буде вивчатися | Основи хмарних технологій. Основні сервіси AWS. |
| Чому це цікаво/треба вивчати | Хмарні технології є однією з найбільш швидкозростаючих галузей в ІТ. Знання AWS є важливим для будь-якого ІТ-фахівця. |
| Чому можна навчитися | Розуміти основні концепції хмарних обчислень. Створювати та управляти хмарною інфраструктурою. Використовувати основні сервіси AWS (EC2, S3, RDS, VPC). Автоматизувати та масштабувати хмарну інфраструктуру. |
| Як можна користуватися набутими знаннями і уміннями | Управління хмарною інфраструктурою. Автоматизація ІТ-процесів. Підвищення рівня власної ІТ-компетентності. Цей курс надасть вам необхідні знання та навички для успішної роботи з хмарними технологіями на платформі AWS. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

8. Cloud Technologies Fundamentals

| Course | Cloud Technologies Fundamentals |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 7 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Computer Systems and Networks Fundamentals". |
| What will be studied | Cloud Fundamentals. AWS Core Services. |
| Why is this interesting / worth exploring | Cloud technologies are one of the fastest growing industries in IT. Knowledge of AWS is essential for any IT professional. |
| What can you learn | Understand the basic concepts of cloud computing. Create and manage cloud infrastructure. Use core AWS services (EC2, S3, RDS, VPC). Automate and scale cloud infrastructure. |
| How to use the acquired knowledge and skills | Manage cloud infrastructure. Automate IT processes. Improve your own IT competency. This course will provide you with the necessary knowledge and skills to successfully work with cloud technologies on the AWS platform. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

9. Обробка та моделювання даних мовою Python

| Дисципліна | Обробка та моделювання даних мовою Python |
|---|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 7 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практик: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями із дисциплін «Основи програмування», «Програмування», «Бази даних». |
| Що буде вивчатися | Курс охоплює інструменти для ефективної обробки та візуалізації великих об'ємів інформації на Python, знайомить студентів з особливостями обробки даних використовуючи сучасні графічні можливості Python. Частина 1. Вступ в Python. Основні конструкції мови Python. Робота з масивами. Графічні можливості Python. Символьні обчислення. Бібліотеки NumPy та SymPy. Частина 2. Створення та використання класів. Розроблення віконних додатків. Модуль Turtle. Бібліотеки Tkinter, Matplotlib та SciPy. |
| Чому це цікаво/треба вивчати | Python — одна з найпростіших мов для вивчення, завдяки її простому синтаксису та читабельності. Він також вимагає значно меншої кількості рядків для написання програм. Для Python існує безліч бібліотек, які можна використовувати на різних етапах обробки даних, більшість з них безкоштовні. Все це впливає на простоту роботи з даними за допомогою Python. |
| Чому можна навчитися | Можна навчитися представляти дані у тому вигляді, в якому вони матимуть сенс, а саме систематизувати та обробляти отримані дані, а потім будувати моделі з використанням бібліотек Python (NumPy, SymPy, Tkinter, Matplotlib та SciPy). |
| Як можна користуватися набутими знаннями і уміннями | Студенти, які закінчать курс, зможуть моментально робити складні розрахунки, проводити статистичний аналіз великих масивів даних та будувати графіки і системи діаграм за кілька рядків коду. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

10. Data Processing and Modeling in Python

| Course | Data Processing and Modeling in Python |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 7 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Fundamentals of Programming", "Programming", "Databases". |
| What will be studied | <p>The course covers tools for efficient processing and visualization of large amounts of information in Python, introduces students to the features of data processing, using the modern graphical capabilities of Python.</p> <p>Part 1. Introduction to Python. Basic constructions of the Python. Working with arrays. Graphical capabilities of Python. Symbolic calculations. NumPy and SymPy libraries.</p> <p>Part 2. Creating and using classes. Development of window applications. Turtle module. Tkinter, Matplotlib and SciPy libraries.</p> |
| Why is this interesting / worth exploring | Python is one of the easiest languages to learn due to its simple syntax and readability. It also requires far fewer lines of code to write programs. There are many libraries for Python that can be used at different stages of data processing, most of them are free. All this affects the ease of working with data using Python. |
| What can you learn | You can learn how to represent data in a way that makes sense, organize and process the received data, and then build models using Python libraries (NumPy, SymPy, Tkinter, Matplotlib and SciPy). |
| How to use the acquired knowledge and skills | Students who complete the course will be able to instantly perform complex calculations, conduct statistical analysis of large datasets, and build graphs and charting systems in a few lines of code. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

11. Бізнес-аналіз процесів автоматизації програмного забезпечення

| Дисципліна | Бізнес-аналіз процесів автоматизації програмного забезпечення |
|---|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 8 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння загальними знаннями із дисциплін комп'ютерної інженерії |
| Що буде вивчатися | Метою викладання дисципліни є засвоєння студентами концепцій та принципів автоматизації в управлінні інформаційними процесами згідно міжнародним стандартом бізнес-аналізу в проектах розробки програмного забезпечення. Формування практичних фахових вмінь бізнес-аналітика процесів автоматизації. |
| Чому це цікаво/треба вивчати | 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"> – процесного моделювання та проектування; – побудови архітектури процесів програмного забезпечення; – бізнес-аналізу інформаційних систем. |
| Чому можна навчитися | Знання: <ul style="list-style-type: none"> – методів та наукових підходів до моделювання процесів; – методів логіко-структурного аналізу; – способів формування концепції системи процесів; – методів проектування процесів; – особливостей застосування міжнародних стандартів управлінню бізнес-процесами. Уміння: <ul style="list-style-type: none"> – визначати вимоги до управління процесами; – моделювати та аналізувати процеси; – проектувати та оптимізувати процеси; – планування дій до автоматизації бізнес-процесів. |
| Як можна користуватися набутими знаннями і уміннями | Засвоєння навчальної дисципліни дозволить набути фахові компетенції: <ul style="list-style-type: none"> – комунікації з замовниками програмного забезпечення; – збір вимог до процесів автоматизації; – моделювання та аналіз процесів автоматизації; – проектування процесів автоматизації; – управління процесами ефективної автоматизації; – обирати інформаційну технологію відповідно до визначених вимог до інформаційної системи. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

12. Business Analysis of Software Automation Processes

| Course | Business Analysis of Software Automation Processes |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 8 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Possession of general knowledge of the computer engineering disciplines |
| What will be studied | <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> |
| Why is this interesting / worth exploring | <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"> – process modeling and design; – of building architecture of software process; – business analysis of information systems. |
| What can you learn | <p>To know:</p> <ul style="list-style-type: none"> – methods and scientific approaches to process modeling; – methods of logical-structural analysis; – techniques to form concepts of process system; – methods of process design; – features of applying international standards for business process management <p>Be able to:</p> <ul style="list-style-type: none"> – define requirements to process management; – model and to analyzer processes; – design and optimize processes; – plane actions to automation business processes |
| How to use the acquired knowledge and skills | <p>Be able to:</p> <ul style="list-style-type: none"> – communicate with customers of software; – collect requirements to automation processes; – model and analyzer automation processes; – design automation processes; – manager effective automation processes; – choose information technology according to defined requirements of information system. |
| Information support of the course | <p>Syllabus</p> <p>Electronic lecture notes</p> |
| Semester assessment | Final Test |

13. Інфраструктура програмного забезпечення

| Дисципліна | Інфраструктура програмного забезпечення |
|---|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 8 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями з навчальних дисциплін «Організація комп'ютерних систем і мереж», «Основи програмування», «Програмування», «Бази даних». |
| Що буде вивчатися | 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). |
| Чому це цікаво/треба вивчати | Метою дисципліни є набуття теоретичних знань і практичних навиків розроблення контейнерного програмного забезпечення. Використання набутих практичних навиків дозволить проєктувати, розробляти та експлуатувати контейнерне програмне забезпечення. |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – основи програмування інфраструктури (інфраструктура як код); – архітектуру контейнерного програмного забезпечення; – життєвий цикл розроблення контейнерного програмного забезпечення; – основи контейнеризування і оркестрування програмного забезпечення (Docker, Kubernetes); – основи керування конфігуруванням програмного забезпечення (Ansible, Chef, Salt, Puppet); – основи забезпечення інфраструктури (Terraform, Cloud formation). |
| Як можна користуватися набутими знаннями і уміннями | Уміти: <ul style="list-style-type: none"> – специфікувати вимоги до контейнерного програмного забезпечення; – створювати статичну логічну структуру контейнерного програмного забезпечення; – створювати динамічну логічну структуру контейнерного програмного забезпечення; – створювати фізичну структуру контейнерного програмного забезпечення; – програмно реалізовувати контейнерного програмного забезпечення. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

14. Software Infrastructure

| Course | Software Infrastructure |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 8 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Fundamentals of Computer Systems and Networks", "Fundamentals of Programming", "Programming", "Databases". |
| What will be studied | 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). |
| Why is this interesting / worth exploring | 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. |
| What can you learn | To know: <ul style="list-style-type: none"> – basics of infrastructure programming (infrastructure as code); – architecture of container software; – life cycle of developing container software; – basics of software containerization and orchestration (Docker, Kubernetes); – basics of software configuration management (Ansible, Chef, Salt, Puppet); – basics of infrastructure provision (Terraform, Cloud formation). |
| How to use the acquired knowledge and skills | Be able to: <ul style="list-style-type: none"> – specify requirements to container software; – develop static logic structure of container software; – develop dynamic logic structure of container software; – develop physical structure of container software; – implement of container software. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

15. Програмні засоби захисту інформації

| Дисципліна | Програмні засоби захисту інформації |
|------------------------------|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 8 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практик: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Досвід командного розроблення програмного забезпечення. |
| Що буде вивчатися | Класифікація криптографічних систем, елементи криптоаналізу, сфери застосування криптографії, моноалфавітні шифри заміни, поліалфавітні шифри заміни, шифри перестановки, шифр Фейстеля, стандарт шифрування даних (DES), стандарт AES, міжнародний алгоритм шифрування даних (IDEA), Blowfish, RC5, арифметика у класах лишків за модулем, теорема Ферма та Ейлера, алгоритми перевірки чисел на простоту, арифметика довгих чисел, генерування випадкових простих чисел, принципи побудови криптосистем з відкритим ключем, обмін ключами за схемою Діффі-Хелмана, Алгоритм RSA та методи факторизації чисел, схема Ель-Гамала та методи знаходження дискретного логарифму, визначення еліптичної кривої, еліптичні криві над скінченними полями $GF(p)$ та $GF(p^m)$, еліптичний аналог обміну ключами за схемою Діффі-Хелмана, алгоритми множення точки ЕК на число, задача дискретного логарифмування на еліптичній кривій, еліптичний аналог алгоритму RSA та схеми Ель-Гамала, принципи побудови криптостійких еліптичних кривих, стандарт цифрового підпису (DSS), алгоритм RSA та схема Ель-Гамала в режимі цифрового підпису, еліптичні алгоритми формування електронного цифрового підпису. |
| Чому це цікаво/треба вивчати | Після вивчення даної дисципліни у студентів формуються здатності: <ul style="list-style-type: none"> – аналізувати криптографічні системи; – обирати криптографічний алгоритм відповідно до сформульованої задачі; – забезпечувати роботу сучасних криптосистем; – виконувати налагодження та розробку програмного забезпечення для криптографічного захисту. |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – методи захисту від несанкціонованого читання, контролю цілісності інформації, аутентифікації, захисту документів та цінних паперів від підробок; – способи виконання операцій у скінченному полі виду $GF(p)$ та його розширенні $GF(p^m)$, а також на еліптичній кривій; – алгоритми розподілу ключів, формування цифрового підпису, шифрування даних; – принципи розробки симетричних та асиметричних криптографічних алгоритмів; – методики застосування сучасних криптографічних алгоритмів; – особливості застосування криптографічних алгоритмів побудованих на еліптичній кривій. |

| | |
|--|---|
| | <p>Уміти:</p> <ul style="list-style-type: none"> – аналізувати результати роботи відомих криптографічних протоколів; – оцінювати стійкість криптографічних алгоритмів; – узагальнювати отримані експериментальні результати. |
| Як можна користуватися набутими знаннями і уміннями | <ul style="list-style-type: none"> – проєктування систем криптографічного захисту; – розроблення програмних засобів шифрування, дешифрування даних та створення електронного цифрового підпису; – створення програмного забезпечення еліптичної криптографії; – застосування криптографічних стандартів при розробленні програмного забезпечення. |
| Інформаційне забезпечення | <p>Силабус Електронний конспект лекцій</p> |
| Семестровий контроль | Залік |

16. Software Instruments of Information Protection

| Course | Software Instruments of Information Protection |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 8 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Experience in team development of software. |
| What will be studied | 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. |
| Why is this interesting / worth exploring | After studying this discipline students will develop the ability to: <ul style="list-style-type: none"> – analyze cryptographic systems; – select cryptographic algorithm according to formulated task; – provision operation of modern cryptosystems; – debug and develop software for cryptographic protection. |
| What can you learn | To know: <ul style="list-style-type: none"> – protection methods against unauthorized reading, control methods of information integrity, authentication methods, protection methods of documents and securities from forgeries; – methods of performing operations in finite field of form $GF(p)$ and its extension $GF(p^m)$, as well as on elliptic curve; – algorithms of key distribution, digital signature generation, data encryption; – principles of developing symmetric and asymmetric cryptographic algorithm; – methods of applying modern cryptographic algorithms; – features of applying cryptographic algorithms based on elliptic curve. Be able to: <ul style="list-style-type: none"> – analyze work results of known cryptographic protocols; – estimate the stability of cryptographic algorithms; |

| | |
|---|--|
| | – summarize the obtained experimental results. |
| How to use the acquired knowledge and skills | Be able to: <ul style="list-style-type: none"> – 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. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

17. Технологія Block Chain

| Дисципліна | Технологія Block Chain |
|--|--|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 8 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практ: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями з навчальних дисциплін «Організація комп'ютерних систем і мереж», «Основи програмування», «Програмування», «Бази даних», «Компоненти програмної інженерії». |
| Що буде вивчатися | 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. |
| Чому це цікаво/треба вивчати | 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 розподіленні програмні застосунки. |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – основи використання технології блокчейн; – переваги та недоліки використання технології блокчейн; – архітектуру 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. |
| Як можна користуватися набутими знаннями і уміннями | Уміти: <ul style="list-style-type: none"> – обирати та використовувати програмні застосунки технології блокчейн; – проектувати розподілені програмні застосунки; – розробляти розподілені програмні застосунки; – експлуатувати розподілені програмні застосунки. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

18. Block Chain Technology

| Course | Block Chain Technology |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 8 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Fundamentals of Computer Systems and Networks", "Fundamentals of Programming", "Programming", "Databases", "Components of Software Engineering". |
| What will be studied | 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 |
| Why is this interesting / worth exploring | 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. |
| What can you learn | To know: <ul style="list-style-type: none"> – fundamentals of using blockchain technology; – advantages and disadvantages of using blockchain technology; – blockchain software architecture; – ways to determine 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 the Ethereum platform. |
| How to use the acquired knowledge and skills | To be able: <ul style="list-style-type: none"> – select and use blockchain software applications; – design distributed software applications; – develop distributed software applications; – exploit distributed software applications. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

19. Емпіричні методи програмної інженерії

| Дисципліна | Емпіричні методи програмної інженерії |
|--|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 8 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практик: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями із дисциплін «Алгоритми та структури даних», «Програмування», «Математичний аналіз», «Лінійна алгебра та аналітична геометрія». Досвід розроблення програмного забезпечення на будь-якій мові програмування |
| Що буде вивчатися | 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. |
| Чому це цікаво/треба вивчати | Числа, які обираються випадковим чином, знаходять багато корисних застосувань: моделювання, програмування, прийняття рішень, математика та розваги. При цьому від якості генераторів випадкових чисел на пряму 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". |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – методи генерування псевдовипадкових чисел з рівномірним законом розподілу на заданому інтервалі; – методи генерування псевдовипадкових чисел з похідними від рівномірного законами розподілу із заданими параметрами; – емпіричні критерії оцінки ступеня випадковості послідовності чисел. Уміти: <ul style="list-style-type: none"> – застосовувати на практиці методи генерування цілих псевдовипадкових чисел із заданим законом розподілу та заданими параметрами; – застосовувати на практиці методи генерування дійсних псевдовипадкових чисел із заданим законом розподілу та заданими параметрами; – застосовувати на практиці емпіричні критерії оцінки випадковості послідовності чисел; – розробляти програмне забезпечення для генерування послідовності псевдовипадкових чисел із заданим законом розподілу та заданими параметрами; – розробляти програмне забезпечення для оцінки випадковості послідовності чисел. |
| Як можна користуватися набутими знаннями і уміннями | <ul style="list-style-type: none"> – проектування генераторів псевдовипадкових послідовностей чисел з рівномірним розподілом; – проектування генераторів псевдовипадкових послідовностей чисел з довільним законом розподілу; |

| | |
|----------------------------------|---|
| | – розроблення програмного забезпечення для оцінки випадковості послідовності чисел. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

20. Empirical Software Engineering Methods

| Course | Empirical Software Engineering Methods |
|---|---|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 8 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Algorithms and Data Structures", "Programming", "Higher Mathematics", "Linear Algebra and Analytic Geometry". Experience in software development for any programming language. |
| What will be studied | 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. |
| Why is this interesting / worth exploring | 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". |
| What can you learn | <p>To know:</p> <ul style="list-style-type: none"> – methods of generating pseudo random numbers with uniform distribution law at a given interval; – methods of generating pseudo random numbers with derivatives from uniform distribution laws with given parameters; – empirical criteria of estimating degree of number sequence randomness. <p>Be able to:</p> <ul style="list-style-type: none"> – apply methods of generating pseudo random integer numbers with given distribution law and given parameters in practice; – apply methods of generating pseudo random real numbers with given distribution law and given parameters in practice; – apply empirical criteria of estimating randomness of number sequence in practice; – develop software for generating sequence of pseudo random numbers with given distribution law and given parameters; – develop software for estimating randomness of number sequence. |
| How to use the acquired knowledge and skills | <p>Be able to:</p> <ul style="list-style-type: none"> – design pseudo random number generators with uniform distribution; – design pseudo random number generators with arbitrary distribution law; – develop software to estimate random of number sequence. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |

21. Методи побудови програмних засобів для матричних обчислень

| Дисципліна | Методи побудови програмних засобів для матричних обчислень |
|---|---|
| Кафедра | Програмного забезпечення комп'ютерних систем |
| Рівень ВО | Перший (бакалаврський) |
| Курс, семестр | 4 курс, 8 семестр |
| Обсяг дисципліни | 4 кредити ЄКТС (лекції: 36 год., практик: 18 год., СРС: 66 год.) |
| Мова викладання | Українська |
| Вимоги до початку вивчення | Володіння знаннями з навчальних дисциплін «Математичний аналіз», «Лінійна алгебра та аналітична геометрія», «Алгоритми та структури даних», «Програмування». Досвід командного розроблення програмного забезпечення. |
| Що буде вивчатися | Сучасні методи та програмні засоби для роботи з матрицями великої розмірності та їх місце в математичному моделюванні. |
| Чому це цікаво/треба вивчати | Після вивчення даної дисципліни у студентів формуються здатності: <ul style="list-style-type: none"> – аналізувати програмні методи виконання матричних обчислень; – аналізувати та оцінювати похибки проведених обчислень над матрицями великої розмірності; – виконувати вибір необхідних математичних та програмних функцій для розв'язання поставленої задачі; – розробляти та ефективно застосувати програмні додатки для чисельного розв'язання матричних задач. |
| Чому можна навчитися | Знати: <ul style="list-style-type: none"> – методи розв'язання часткової проблеми власних значень; – методи розв'язання повної проблеми власних значень; – метод виконання сингулярного розкладення прямокутної матриці; – методи апроксимації таблично-заданих функцій; – методи програмного розв'язання систем нелінійних рівнянь. Уміти: <ul style="list-style-type: none"> – аналізувати та оцінювати похибку програмно отриманого розв'язку математичної задачі; – узагальнювати та аналізувати програмні методи розв'язання систем нелінійних рівнянь; – застосовувати на практиці методи розв'язання математичних задач на ЕОМ; – розробляти програмне забезпечення для чисельного розв'язання матричних задач. |
| Як можна користуватися набутими знаннями і уміннями | Уміти: <ul style="list-style-type: none"> – проектувати системи комп'ютерної математики для виконання матричних обчислень; – використовувати сучасні математичні пакети. |
| Інформаційне забезпечення | Силабус Електронний конспект лекцій |
| Семестровий контроль | Залік |

22. Matrix Calculus Software Methods

| Course | Matrix Calculus Software Methods |
|---|--|
| Department | Computer Systems Software Department |
| Level of higher education | First (Bachelor) |
| Year of study, semester | 4 th year, 8 th semester |
| Course total scope | 4 credits (Lectures: 36 academic hours, computer class: 18 academic hours, student's self-training: 66 academic hours). |
| Language of study | English |
| Requirements for begin studying the course | Knowledge of the disciplines "Higher Mathematics", "Algorithms and Data Structures", "Linear Algebra and Analytic Geometry", "Programming". Experience in team software development. |
| What will be studied | Modern methods and software for working with big dimensional matrices and their place in mathematical modeling. |
| Why is this interesting / worth exploring | <p>After studying this discipline, students be able to:</p> <ul style="list-style-type: none"> – to analyze software methods of performing matrix calculations; – to analyze and estimate the errors of the calculations on the matrices of big dimensions; – to perform the selection of the necessary mathematical and software functions to solve the problem; – to develop and effectively apply software applications for numerical solution of matrix problems. |
| What can you learn | <p>To know:</p> <ul style="list-style-type: none"> – methods of solving a partial problem of eigenvalues; – methods of solving the complete problem of eigenvalues; – method of performing singular decomposition of a rectangular matrix; – methods of approximation of tabular - given functions; – software methods for solving of systems of nonlinear equations. <p>Be able to:</p> <ul style="list-style-type: none"> – analyze and estimate the error of the software obtained solution of the mathematical problem; – generalize and analyze software methods for solving systems of nonlinear equations; – apply in practice methods of solving mathematical problems on a computer; – develop software for numerical solution of matrix problems. |
| How to use the acquired knowledge and skills | <p>Be able to:</p> <ul style="list-style-type: none"> – design of computer mathematics systems for performing matrix calculations; – use of modern mathematical packages. |
| Information support of the course | Syllabus Electronic lecture notes |
| Semester assessment | Final Test |