

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1. Загальна інформація про навчальну дисципліну

| | |
|--|--|
| Повна назва навчальної дисципліни | Сучасні |
| Повна офіційна назва закладу вищої освіти | Сумський державний університет |
| Повна назва структурного підрозділу | Факультет електроніки та інформаційних технологій. Кафедра комп'ютерних наук |
| Розробник(и) | Москаленко Альона Сергіївна |
| Рівень вищої освіти | Перший рівень вищої освіти, НРК – 6 рівень, QF-LLL – 6 рівень, FQ-EHEA – перший цикл |
| Семестр вивчення навчальної дисципліни | 16 тижнів протягом 4-го семестру |
| Обсяг навчальної дисципліни | Обсяг становить 5 кред. ЄКТС, 150 год. Для денної форми навчання 64 год. становить контактна робота з викладачем (32 год. лекцій, 32 год. лабораторних занять), 86 год. становить самостійна робота. |
| Мова викладання | Українська |

2. Місце навчальної дисципліни в освітній програмі

| | |
|---|---|
| Статус дисципліни | Обов'язкова навчальна дисципліна для освітньої програми "Інформатика" |
| Передумови для вивчення дисципліни | Програмування, Алгоритми і структури даних |
| Додаткові умови | Додаткові умови відсутні |
| Обмеження | Обмеження відсутні |

3. Мета навчальної дисципліни

Полягає у забезпеченні студентів комплексним розумінням різних підходів та парадигм програмування, що є фундаментальними для розробки сучасних програмного забезпечення. Цей курс служить як ключовий елемент, що з'єднує базові знання з програмування, отримані студентами, з більш складними та спеціалізованими концептами, необхідними для розробки технологічних рішень.

4. Зміст навчальної дисципліни

Тема 1 Основні парадигми програмування та їх застосування

Загальний огляд парадигм програмування Короткий історичний огляд розвитку парадигм. Імперативне програмування Декларативне програмування Процедурне програмування Об'єктно-орієнтоване програмування (ООП) Основні концепції ООП: класи, об'єкти, спадкування, інкапсуляція. Основні принципи функціонального програмування. Логічне програмування Введення в функціональне програмування Чисті функції Визначення чистих функцій. Переваги використання чистих функцій у програмуванні. Незмінність даних Функції вищого порядку Паралельні обчислення у функціональному програмуванні Як функціональне програмування сприяє ефективному паралельному виконанню. Приклади використання паралельних обчислень. Функтори та їх застосування Монади в функціональному програмуванні Сучасне застосування функціонального програмування Огляд сучасних мов програмування, що підтримують функціональний стиль.

Тема 2 C++: особливості та практичне застосування

Вступ до C++: Короткий огляд нововведень C++. Перевантаження функцій: Як і чому використовувати. Імена-етикетки: Їх роль та використання. Inline-функції: Переваги та приклади використання. Визначення дружніх функцій та класів: Огляд концепції. Друзі в C++: дружні функції та класи Перевантажені операції в C++ Статичні елементи, константні методи та анонімні об'єкти в C++ Наслідування класів та модифікатори доступу в C++ Віртуальні функції та базові класи в C++ Абстрактні класи та чисті віртуальні функції в C++ Робота з файлами та потоками в C++ Шаблони функцій та класів в C++

5. Очікувані результати навчання навчальної дисципліни

Після успішного вивчення навчальної дисципліни здобувач вищої освіти зможе:

| | |
|-----|--|
| PH1 | Компілювати, збирати і виконувати проекти на C++. Застосовувати C++ Code Conventions. |
| PH2 | Описувати ієрархію наслідування класів і уникати двозначності при множинному наслідуванні. |
| PH3 | Перевантажувати операції та методи для класів. |
| PH4 | Використовувати стандартну бібліотеку шаблонів STL. |
| PH5 | Описувати прикладну задачу в сутностях об'єктно-орієнтованої парадигми. |

6. Роль навчальної дисципліни у досягненні програмних результатів

Програмні результати навчання, досягнення яких забезпечує навчальна дисципліна.

Для спеціальності 122 Комп'ютерні науки:

| | |
|-----|--|
| PR5 | Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій. |
| PR9 | Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук. |

| | |
|------|--|
| ПР15 | Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем. |
| ПР18 | Застосовувати знання сучасних технологій, парадигм та мов програмування, платформ реалізації, методів і стандартів розробки програмного забезпечення. |

7. Роль освітнього компонента у формуванні соціальних навичок

Загальні компетентності та соціальні навички, формування яких забезпечує навчальна дисципліна:

| | |
|-----|--|
| СН1 | Знання та розуміння предметної області та розуміння професійної діяльності |
| СН2 | Здатність вчитися й оволодівати сучасними знаннями |

8. Види навчальних занять

| | |
|---|--|
| Тема 1. Основні парадигми програмування та їх застосування | |
| Лк1 "Введення в сучасні парадигми програмування" (денна) Загальний огляд парадигм програмування Вступ та значення парадигм у програмуванні. Короткий історичний огляд розвитку парадигм. Імперативне програмування Визначення та ключові характеристики. Як імперативне програмування впливає на підхід до розробки ПЗ. Декларативне програмування Розрізнення між декларативним та імперативним стилями. Приклади декларативного програмування у практиці. Процедурне програмування Визначення та приклади. Застосування процедурного програмування у різних областях. Об'єктно-орієнтоване програмування (ООП) Основні концепції ООП: класи, об'єкти, спадкування, інкапсуляція. Переваги та недоліки ООП. Основні принципи функціонального програмування. Як функціональне програмування вирішує певні проблеми в програмуванні. Логічне програмування Основи логічного програмування. Приклади застосування логічного програмування. | |
| Лк2 "Структурне програмування: Поняття та особливості" (денна) Введення у структурне програмування Визначення структурного програмування. Історичний контекст та розвиток структурного програмування. Основні концепції структурного програмування Структуризація програм та коду. Використання контрольних структур: послідовність, вибір, цикл. Порівняння структурного та процедурного програмування Схожості та відмінності між структурним та процедурним підходами. Переваги використання структурного програмування над процедурним. Структурне програмування у сучасних мовах Як структурне програмування вплинуло на сучасні мови програмування. Приклади структурного програмування на мовах С, С++, Java. Практичне застосування структурного програмування Типові випадки використання структурного підходу у програмуванні. Розробка алгоритмів з використанням структурного стилю. | |

Лк3 "Процедурне програмування: Основи та застосування" (денна)

Визначення процедурного програмування Основні концепції та визначення. Історичний контекст та еволюція процедурного програмування. Основні характеристики процедурного програмування Структура програм у процедурному стилі. Використання функцій та процедур. Область застосування процедурного програмування Типові сценарії використання. Переваги процедурного підходу у певних типах задач. Переваги процедурного програмування Чому процедурний підхід ефективний у певних ситуаціях. Порівняння процедурного програмування з іншими парадигмами. Недоліки процедурного програмування Обмеження та потенційні проблеми процедурного підходу. Коли краще вибирати інші парадигми. Практичні приклади процедурного програмування Демонстрація типових прикладів коду. Аналіз використання процедурного стилю у сучасних програмах.

Лк4 "Об'єктно-орієнтоване програмування: Класичний підхід" (денна)

Основи об'єктно-орієнтованого програмування (ООП) Визначення та ключові принципи ООП. Історичний контекст та розвиток ООП. Класи та об'єкти Визначення класів та об'єктів. Створення класів та інстанціювання об'єктів. Успадкування та Ієрархія класів Принципи успадкування у ООП. Створення ієрархій класів та їх використання. Інкапсуляція та Абстракція Захист даних та методів в класах. Визначення абстрактних класів та інтерфейсів. Поліморфізм Визначення поліморфізму та його види у ООП. Приклади застосування поліморфізму. Практичне застосування ООП. Реальні приклади використання ООП у програмуванні. Переваги та недоліки підходу ООП. CASE-засоби для моделювання класів та об'єктів

Лк5 "Основи функціонального програмування" (денна)

Введення в функціональне програмування Означення функціонального програмування. Історичний контекст та філософія функціонального програмування. Чисті функції Визначення чистих функцій. Переваги використання чистих функцій у програмуванні. Незмінність даних Концепція незмінності даних. Як незмінність впливає на побудову програм. Функції вищого порядку Визначення та приклади функцій вищого порядку. Використання функцій вищого порядку у практичному програмуванні. Переваги та виклики функціонального програмування Обговорення сценаріїв, де функціональне програмування є найбільш ефективним. Обговорення викликів, пов'язаних із функціональним програмуванням. Використання функціональних мов у системному проектуванні.

Лк6 "Принципи функціонального програмування" (денна)

Паралельні обчислення у функціональному програмуванні Як функціональне програмування сприяє ефективному паралельному виконанню. Приклади використання паралельних обчислень. Доведення коректності програм Методи та підходи до доведення коректності програм у функціональному стилі. Важливість формальних доказів у функціональному програмуванні. Функтори та їх застосування Визначення функторів у контексті функціонального програмування. Приклади використання функторів у програмуванні. Монади в функціональному програмуванні Роз'яснення концепції монад та їх ролі. Приклади використання монад у реальних програмах. Сучасне застосування функціонального програмування Огляд сучасних мов програмування, що підтримують функціональний стиль. Приклади сучасних програмних систем, побудованих на принципах функціонального програмування.

| |
|---|
| <p>Лб1 "Вибір мови програмування та встановлення середовища розробки" (денна) Ознайомлення з різними мовами програмування, вибір мови для курсу. Застосування C++ Code Conventions</p> |
| <p>Лб2 "Застосування процедурного програмування на C++" (денна) Реалізація основних конструкцій мови C++.</p> |
| <p>Лб3 "Структурне програмування на практиці в C++" (денна) Використання структурних конструкцій мови C++.</p> |
| <p>Лб4 "ООП в C++: Класи та об'єкти" (денна) Створення та використання класів та об'єктів. Використання CASE-інструментів для моделювання класів та об'єктів.</p> |
| <p>Лб5 "Функціональне програмування: Основи" (денна) Реалізація чистих функцій, функцій вищого порядку. Розробка функціональних моделей для задач системного аналізу.</p> |
| <p>Лб6 "Застосування принципів функціонального програмування" (денна) Практичне впровадження функціонального програмування.</p> |
| <p>Тема 2. C++: особливості та практичне застосування</p> |
| <p>Лк7 "Відмінності мови C++ від C" (денна) Вступ до C++: Короткий огляд нововведень C++. Передача даних: Порівняння методів передачі даних у C++ та C. Посилання в C++: Введення та використання посилань у C++.</p> |
| <p>Лк8 "Перевантажені операції." (денна) Перевантаження функцій та вбудовані функції в C++. Перевантаження функцій: Як і чому використовувати. Імена-етикетки: Їх роль та використання. Inline-функції: Переваги та приклади використання.</p> |
| <p>Лк9 "Друзі в C++: дружні функції та класи" (денна) Визначення дружніх функцій та класів: Огляд концепції. Доступ до приватних і захищених членів. Обмеження та переваги: Аналіз сильних та слабких сторін використання дружніх функцій. Застосування: Практичні приклади та випадки використання.</p> |
| <p>Лк10 "Перевантажені операції в C++" (денна) Вивчення перевантажених операцій. Основи перевантаження операцій : Як і чому їх використовувати. Приклади перевантажених операцій : Розгляд конкретних випадків.</p> |

| |
|--|
| <p>Лк11 "Статичні елементи, константні методи та анонімні об'єкти в C++" (денна)</p> <p>Розгляд статичних елементів, константних об'єктів і методів, анонімних об'єктів класу. Статичні елементи : Їх роль та використання. Константні методи та об'єкти : Визначення та застосування.</p> |
| <p>Лк12 "Наслідування класів та модифікатори доступу в C++" (денна)</p> <p>Призначення, описання, приклади наслідування класів, модифікатори доступу. Наслідування класів : Основи та приклади. Модифікатори доступу : Public, Private, Protected</p> |
| <p>Лк13 "Віртуальні функції та базові класи в C++" (денна)</p> <p>Віртуальні функції, базові класи. Віртуальні функції : Введення та застосування. Базові класи : Як і чому їх використовувати у віртуальних функціях.</p> |
| <p>Лк14 "Абстрактні класи та чисті віртуальні функції в C++" (денна)</p> <p>Абстрактні класи, чисті віртуальні функції. Абстрактні класи : Огляд та приклади. Чисті віртуальні функції : Їх роль у абстрактних класах.</p> |
| <p>Лк15 "Робота з файлами та потоками в C++" (денна)</p> <p>Конструктори файлових потоків, форматування даних. Конструктори файлових потоків : Використання для читання та запису. Форматування даних : Методи та приклади.</p> |
| <p>Лк16 "Шаблони функцій та класів в C++" (денна)</p> <p>Шаблони функцій, класів. Шаблони функцій : Огляд та застосування. Шаблони класів : Приклади та вигоди використання. Застосування шаблонів у проектуванні програмного забезпечення.</p> |
| <p>Лб7 "Робота з особливостями мови C++" (денна)</p> <p>Використання специфічних особливостей мови C++. Функції та область видимості</p> |
| <p>Лб8 "Перевантаження функцій та вбудовані функції в практиці" (денна)</p> <p>Перевантаження функцій. Створення декількох функцій з однаковою назвою, але різними параметрами. Розгляд правил та обмежень перевантаження функцій. Практичні приклади з різними типами параметрів та кількістю аргументів. Використання inline-функцій. Оцінка переваг та потенційних недоліків використання перевантажених та inline-функцій.</p> |
| <p>Лб9 "Дружні функції та класи в C++: практика" (денна)</p> <p>Реалізація та використання дружніх функцій та класів.</p> |
| <p>Лб10 "Практика з перевантаженими операціями в C++" (денна)</p> <p>Розв'язання задач з перевантаженими операціями.</p> |
| <p>Лб11 "Робота зі статичними елементами та константними методами" (денна)</p> <p>Практичне використання статичних елементів, константних методів.</p> |

| |
|--|
| Лб12 "Практика наслідування класів в С++" (денна) Розв'язування задач із застосуванням наслідування класів. |
| Лб13 "Робота з віртуальними функціями та базовими класами" (денна) Реалізація та використання віртуальних функцій. |
| Лб14 "Застосування абстрактних класів та чистих віртуальних функцій" (денна) Створення абстрактних класів, робота з чистими віртуальними функціями. |
| Лб15 "Робота з файлами та потоками в С++: практика" (денна) Використання файлових потоків, форматування даних. |
| Лб16 "Практика застосування шаблонів в С++" (денна) Створення та використання шаблонів функцій і класів. Проектування шаблонів класів та функцій у контексті розробки систем. |

9. Стратегія викладання та навчання

9.1 Методи викладання та навчання

Дисципліна передбачає навчання через:

| | |
|-----|------------------------------|
| МН1 | Електронне навчання |
| МН2 | Лекційне навчання |
| МН3 | Самостійне навчання |
| МН4 | Практикоорієнтоване навчання |

Дисципліна передбачає навчання через: МН 1 Електронне навчання є основою усіх видів навчальної діяльності здобувачів на єдиній навчальній платформі. МН 2 Лекційне навчання – пояснювально-ілюстративний матеріал із залученням студента до дискусії, шляхом інтерактивного способу проведення лекції; МН 3 Випереджувальна самостійна робота з метою вивчення здобувачами нового матеріалу до його представлення на лекції занять. Самостійне виконання лабораторних робіт згідно рекомендацій. Засвоєння тем, що виносяться на самостійне опрацювання (надається на лекції). МН4 Практикоорієнтоване навчання, яке сприяє закріпленню вивченого матеріалу шляхом його використання при виконанні лабораторних робіт.

Навички самонавчання, вміння швидко оцінювати і виділяти потрібну інформацію, критично мислити, порівнюючи прочитаний матеріал. Розвиток логічного та алгоритмічного мислення.

9.2 Види навчальної діяльності

| | |
|-----|--|
| НД1 | Електронне навчання у системах (Google Classroom, MIX) |
| НД2 | Лекції-дискусії |
| НД3 | Підготовка до поточного та підсумкового контролю |
| НД4 | Робота з підручниками та релевантними інформаційними джерелами |

| | |
|-----|--|
| НД5 | Виконання практичних завдань на лабораторних роботах |
|-----|--|

10. Методи та критерії оцінювання

10.1. Критерії оцінювання

| Визначення | Чотирибальна національна шкала оцінювання | Рейтингова бальна шкала оцінювання |
|---|---|------------------------------------|
| Відмінне виконання лише з незначною кількістю помилок | 5 (відмінно) | $90 \leq RD \leq 100$ |
| Вище середнього рівня з кількома помилками | 4 (добре) | $82 \leq RD < 89$ |
| Загалом правильна робота з певною кількістю помилок | 4 (добре) | $74 \leq RD < 81$ |
| Непогано, але зі значною кількістю недоліків | 3 (задовільно) | $64 \leq RD < 73$ |
| Виконання задовольняє мінімальним критеріям | 3 (задовільно) | $60 \leq RD < 63$ |
| Можливе повторне складання | 2 (незадовільно) | $35 \leq RD < 59$ |
| Необхідний повторний курс з навчальної дисципліни | 2 (незадовільно) | $0 \leq RD < 34$ |

10.2 Методи поточного формативного оцінювання

| | Характеристика | Дедлайн, тижні | Зворотний зв'язок |
|--|--|------------------------------------|-------------------------------|
| МФО1 Обговорення та самокорекція виконаної роботи студентами | Обговорення результатів лабораторних робіт, теоретичних положень з метою уточнення та пояснення навчального матеріалу. | кожна лабораторна робота та лекція | google meet, google classroom |
| МФО2 Перевірка та аналіз письмових звітів | Письмові коментарі викладача за результатами перевірки звітів з лабораторних робіт та подальший їх аналіз на лабораторних заняттях | кожне лабораторне заняття | Google classroom, google meet |
| МФО3 Надання зворотного зв'язку про результати перевірки навчальних досягнень здобувачів за матеріалом, що вивчається | Групові чи індивідуальні консультації згідно розкладу, що оприлюднено на сайті кафедри | Протягом консультації | Google Meet |

10.3 Методи підсумкового сумативного оцінювання

| | Характеристика | Дедлайн, тижні | Зворотний зв'язок |
|--|--|---|---|
| МСО1 Підсумковий контроль: екзамен | Письмовий екзамен у формі тестування | Екзаменаційний тиждень | університетська платформа електронного навчання MIX |
| МСО2 Звіт за результатами виконання лабораторних робіт | Звіт за результатами виконання лабораторної роботи повинен бути оформлений відповідно шаблону, наданого лектором курсу, і містити детальний опис результату, отриманого на кожному з етапів виконання. | кожне друге лабораторне заняття | google meet, google classroom |
| МСО3 Поточні контрольні роботи (проміжний модульний контроль) | Проміжний модульний контроль призначений для перевірки рівня засвоєння теоретичного матеріалу. | кожен атестаційний тиждень (9й тиждень) | google meet, google classroom |

Контрольні заходи:

| | | Максимальна кількість балів | Мінімальна кількість балів | Можливість перескладання з метою підвищення оцінки |
|--|------|-----------------------------|----------------------------|--|
| 4 семестр | | 100 балів | | |
| МСО1. Підсумковий контроль: екзамен | | 40 | | |
| | | 40 | Не передбачено | Ні |
| МСО2. Звіт за результатами виконання лабораторних робіт | | 36 | | |
| | 12x2 | 24 | Не передбачено | Ні |
| | 2x6 | 12 | Не передбачено | Ні |
| МСО3. Поточні контрольні роботи (проміжний модульний контроль) | | 24 | | |
| | 2x12 | 24 | Не передбачено | Ні |

Рейтингові бали шкали оцінювання з навчальної дисципліни розподіляються між модульними атестаціями та іспитом відповідно 60 та 40 балів. Іспит проводиться в період екзаменаційної сесії; При отриманні за наслідками модульних атестацій та складання іспиту загального рейтингового балу, що відповідає незадовільній оцінці FX (від 35 до 59 балів), студентові надається право на дворазове складання (викладачеві та комісії) заходу підсумкового семестрового контролю (ПСК), яке здійснюється після завершення останнього модульно-атестаційного циклу у семестрі або екзаменаційної сесії, якщо вона передбачена, за додатковою відомістю семестрової атестації (першою незадовільною оцінкою вважається

та, що отримана за наслідками модульних атестацій та складання іспиту, яка виставляється в основну відомість семестрової атестації). У разі незадовільного складання підсумкового семестрового контролю комісії студент отримує оцінку «незадовільно» («F» за шкалою ECTS) і відраховується з університету. При успішному складанні заходу підсумкового семестрового контролю використовується оцінка «задовільно», яка засвідчує виконання студентом мінімальних вимог без урахування накопичених балів («E» за шкалою ECTS) із визначенням рейтингового балу 60. Студент, який за наслідками модульних атестацій та складання іспиту набрав менше 35 рейтингових балів, не допускається до підсумкового семестрового контролю, отримує оцінку «незадовільно» (за шкалою ECTS – «F») і відраховується з університету.

11. Ресурсне забезпечення навчальної дисципліни

11.1 Засоби навчання

| | |
|-----|--|
| ЗН1 | Бібліотечні фонди |
| ЗН2 | Комп'ютери, комп'ютерні системи та мережи |
| ЗН3 | Програмне забезпечення (Microsoft Visual Studio (2017-2022)) |

11.2 Інформаційне та навчально-методичне забезпечення

| | |
|---|---|
| Основна література | |
| 1 | Brokken F. B. C++ Annotations Version 11.3.0 / F. B. Brokken. // Netherlands: University of Groningen. – 1994-2019. Access: http://www.icce.rug.nl/documents/cplusplus/ |
| Допоміжна література | |
| 2 | Ришковець, Ю. В. Алгоритмізація та програмування [Текст] : навч. посіб. Ч.1 / Ю. В. Ришковець, В. А. Висоцька. – Львів : Новий Світ-2000, 2018. – 337 с. |
| Інформаційні ресурси в Інтернеті | |
| 3 | Комплект електронних навчально методичних матеріалів до дисципліни / Укладач https://mix.sumdu.edu.ua/info/nmk/a33def71-7437-4703-b2de-4124faee7d58 |
| 4 | CS107 - Programming Paradigms https://see.stanford.edu/course/cs107 |
| 5 | Programming Paradigms – Paradigm Examples for Beginners https://www.freecodecamp.org/news/an-introduction-to-programming-paradigms/ |