



**ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ
УНІВЕРСИТЕТ**
Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем

СИЛАБУС (SYLLABUS)
Дисципліна «Алгоритмізація та програмування /
Algorithmization and Programming»

ІНФОРМАЦІЯ ПРО ВИКЛАДАЧА

Викладач	Томашевська Тетяна Володимирівна
Науковий ступінь	Кандидат технічних наук
Вчене звання	Доцент
Посада	Доцент кафедри комп'ютерних наук та інформаційних систем
Адреса кафедри	м. Київ, вул. Кіото 19, каб. Б-507, Б-525
E-mail	compdep@knute.edu.ua
Консультації	Відповідно до графіку індивідуальних консультацій на сайті кафедри

ПОЛІТИКА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

<https://knute.edu.ua/file/MzEyMQ==/c12a9f74e87d9154696ca0f761da2e5c.pdf>

Дотримання академічної доброчесності передбачає:

- самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей);
- посилання на джерела інформації у разі використання не авторських ідей, розробок, тверджень, відомостей і т.п.;
- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної наукової діяльності, використані методики досліджень і джерела інформації.

Порушенням академічної доброчесності вважається:

- академічний плагіат – оприлюднення (частково або повністю) наукових (творчих) результатів, отриманих іншими особами, як результатів власного дослідження (творчості) та/або відтворення опублікованих текстів (оприлюднених творів мистецтва) інших авторів без зазначення авторства;

- самоплагіат – оприлюднення (частково або повністю) власних раніше опублікованих наукових результатів як нових наукових результатів;
- фабрикація – вигадання даних чи фактів, що використовуються в наукових дослідженнях;
- фальсифікація – свідомо зміна чи модифікація вже наявних даних, що стосуються наукових досліджень.

За порушення академічної доброчесності здобувачі освіти можуть бути притягнені до академічної відповідальності:

- повторне проходження оцінювання (модульний контроль, іспит, залік тощо);
- повторне проходження відповідного освітнього компонента освітньо-професійної програми;
- відрахування з Університету;
- позбавлення наданих університетом пільг;
- відмова у присудженні відповідного ступеня вищої освіти;

ПОЛІТИКА ЩОДО ВІДВІДУВАННЯ ЗАНЯТЬ

- відвідування занять є обов'язковим;
- за об'єктивних причин (наприклад, хвороба, міжнародне стажування та ін.) навчання може відбуватись в он-лайн формі за погодженням із викладачем дисципліни.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назва дисципліни / тип дисципліни	Алгоритмізація та програмування / обов'язкова
Навчальний рік	2023-2024
Факультет	Факультет інформаційних технологій
Курс	1
Семестр	1
Освітній ступінь	Бакалавр
Галузь знань	12 «Інформаційні технології»
Спеціальність	126 «Інформаційні системи та технології»
Загальна характеристика	Кількість годин –180 Кількість кредитів – 6 Види занять: лекції, лабораторні, самостійна робота. Співвідношення аудиторних годин і годин самостійної роботи - 102/78 Мова викладання – українська Форма викладання – очна
Підсумковий контроль	Екзамен
Програмне забезпечення	MS Visual Studio 13 та вище

Обладнання	Проектор, комп'ютерна техніка із встановленим програмним забезпеченням та доступом до мережі Інтернет.
Необхідні попередні дисципліни	Шкільний курс «Інформатика»; дисципліна «Дискретна математика»; дисципліна «Теорія інформаційних систем»
Методика вивчення	Методика вивчення дисципліни полягає у набутті студентами знань теоретичного і практично-прикладного характеру під час лекцій, лабораторних занять, самостійної роботи та вивчення першоджерел і навчально-методичної літератури.
Мета і завдання	<p>Метою вивчення дисципліни «Алгоритмізація та програмування» є формування у студентів навичок оволодіння технологіями обробки простих та структурованих даних, опанування технологій структурного, модульного та об'єктно-орієнтованого програмування на базі мови програмування C++.</p> <p>Завданням вивчення дисципліни «Алгоритмізація та програмування» є теоретична та практична підготовка студентів з таких питань: парадигми програмування (структурного, модульного, об'єктно-орієнтованого) та засоби сучасних мов програмування для реалізації різних концепцій; засоби мов програмування для реалізації розгалужених та циклічних алгоритмів; використання функцій та окремих модулів користувача; реалізація класичних інформаційних структур (списків, дерев) з використанням статичного та динамічного розподілу пам'яті; базові технології проектування процесів пошуку та побудови впорядкованих даних у лінійних списках та деревовидних структурах; використання класів для реалізації парадигми об'єктно-орієнтованого програмування; засоби створення ієрархічної об'єктної структури з використанням базового об'єкта, подальше розширення його властивостей з використанням статичних та віртуальних методів; сучасні інструментальні засоби для створення прикладного програмного забезпечення; особливості програмування в операційних системах родини Windows. Система управління повідомленнями. Структура програми для Windows</p>
Місце дисципліни в освітньо-професійній програмі	
Загальні компетентності	КЗ 2 Здатність застосовувати знання у практичних ситуаціях.
Фахові компетентності (результати навчання)	<p>КС 3 Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними.</p> <p>КС 4 Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).</p>
Програмні результати навчання	<p>ПР 2 Застосовувати знання фундаментальних і природничих наук, системного аналізу та технологій моделювання, стандартних алгоритмів та дискретного аналізу при розв'язанні задач проектування і використання інформаційних систем та технологій.</p> <p>ПР 3 Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із</p>

	<p>застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.</p> <p>ПР 4 Проводити системний аналіз об'єктів проектування та обґрунтовувати вибір структури, алгоритмів та способів передачі інформації в інформаційних системах та технологіях.</p> <p>ПР 5 Аргументувати вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.</p> <p>ПР 6 Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.</p> <p>ПР 7 Обґрунтовувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.</p>
--	---

ТЕМАТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

РОЗДІЛ 1. ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ C++

Тема 1.1. Введення у дисципліну «Алгоритмізація та програмування». Огляд можливостей мови програмування C++

Вступ. Мета та завдання дисципліни, її місце у освітньому процесі. Парадигма та основні ідеї, покладені у сучасні алгоритмічні мови програмування. Введення до мови програмування C++, її місце серед інших мов програмування. Види програмного забезпечення для розробки програмних продуктів мовою C++. Знайомство з інструментальними засобами Microsoft Visual Studio. Приклад створення першого проекту для консольної прикладної програми. Основні структурні компоненти програми C++, функція main(). Компіляція та запуск програм на виконання. Етап налагодження програми. Інтерактивні засоби Microsoft Visual C++ для налагодження програм: покрокове виконання програми, перегляд значень змінних, встановлення точок зупинок.

Тема 1.2. Основні поняття мови програмування. Базові типи, константи, змінні, операції, вирази

Алфавіт мови програмування. Поняття лексеми. Лексична структура мови програмування. Типи лексем: константи, літерали, ідентифікатори, ключові слова, операції, розділові знаки. Числові константи у різних системах числення, літерні константи, символні константи, використання спеціальних символів для визначення недрукованих символів. Призначення та використання коментарів у програмі. Поняття синтаксису та семантики мови програмування. Концепція типу даних. Базові типи. Кваліфікатори типів. Сліввідношення між розмірами даних базових типів. Поняття змінної. Оголошення та ініціалізація змінних. Поняття оператора. Простий та складений оператор. Порожній оператор. Оператор присвоєння. Поняття виразу. Операнди та операції. Унарні та бінарні операції. Арифметичні операції. Операції відношення. Логічні операції. Побітові операції. Операції інкременту та декременту. Приклади використання операцій. Пріоритет виконання операцій у виразах. Перетворення типів у виразах.

Тема 1.3. Керування виконанням програми

Поняття розгалужених та циклічних алгоритмів. Приклади задач, розв'язання яких зводиться до застосування розгалужених та циклічних алгоритмів. Рекурентні співвідношення, циклічні ітераційні процеси, табуляція функцій, наближене обчислення функцій. Реалізація розгалужених алгоритмів за допомогою операторів: if ... else ; if ... else if ... else; switch. Порівняння операторів розгалуження. Реалізація циклічних процесів за допомогою операторів: while ; for ; do while. Порівняння операторів циклів. Особливості операторів циклів з лічильником та з умовою. Відмінність між синтаксисом та семантикою циклічних конструкцій з пост- та передумовою. Продовження та достроковий вихід із циклів: оператори break та continue. Оператор переходу goto. Правила побудови вкладених операторів циклів та розгалуження.

Тема 1.4. Складені типи даних: масиви. Алгоритми на масивах

Концепція складених типів даних. Поняття масиву, індексу, елемента. Оголошення та ініціалізація одновимірних масивів. Методи доступу до елементів масивів. Індексний доступ до елементів масиву. Використання вказівників для посилання на елементи масиву. Багатовимірні масиви. Розміщення елементів багатовимірних масивів у пам'яті. Вказівники на багатовимірні масиви. Розрізи багатовимірних масивів. Доступ до рядків та стовпців двовимірних масивів. Використання циклів для обробки масивів. Алгоритм вибору простих чисел за допомогою масиву.

Тема 1.5. Вказівники та адресна арифметика

Поняття та оголошення вказівника. Використання модифікаторів в оголошеннях вказівників. Вказівники типу void. Вказівники та адреси. Операції зі вказівниками. Унарні операції одержання адреси та розкриття посилання вказівника. Динамічний розподіл пам'яті. Операції new, delete. Функції malloc() та free(). Вказівники та аргументи функцій. Використання вказівників для передачі фактичних аргументів за посиланнями. Вказівники на функції. Масиви символів. Використання вказівників для доступу до символів рядків. Масиви вказівників. Ініціалізація масивів вказівників. Вказівники на вказівники. Правила інтерпретації складних оголошень. Приклади складених оголошень.

Тема 1.6. Функції користувача та класи пам'яті

Парадигма структурного та модульного програмування. Загальні відомості про функції користувача. Синтаксис для оголошення та визначення функцій користувача. Формальні та фактичні параметри функцій. Виклик функцій. Підстановка фактичних параметрів за значеннями та посиланнями: основна різниця між двома способами передачі аргументів. Приклади створення функцій користувача та їх виклику. Функції зі змінним числом аргументів. Поняття про рекурсію. Труднощі, що виникають під час реалізації рекурсії. Приклади класичних рекурсивних алгоритмів. Поняття області видимості та "часу життя" програмних об'єктів. Зовнішні змінні та функції. Локальні імена та локальні змінні, автоматична пам'ять. Оголошення та ініціалізація статичних змінних. Регістрові змінні. Змінні класу volatile. Передача аргументів у функцію main().

Тема 1.7. Складені типи даних: структури, об'єднання, перелічення

Концепція складеного типу даних. Основні відомості про структури. Тег та елементи структури. Приклади структур. Описання та ініціалізація структури. Використання декларації typedef для створення нових типів. Операції зі структурами. Доступ до елементів структури за допомогою операції ".". Використання вказівників для одержання доступу до елементів структури: операція "->". Масиви структур. Поняття запису. Рекурсивні оголошення структур з використанням вказівників. Приклади рекурсивних структур: лінійний список, бінарне дерево. Об'єднання (union) як спосіб збереження даних типів даних в одній області пам'яті. Приклади об'єднань. Створення списків-переліків (множин). Декларація enum. Зміна нумерації елементів множин.

РОЗДІЛ 2. C++ ТА ОБ'ЄКТНООРІЄНТОВАНЕ ПРОГРАМУ ВАННЯ

Тема 2.1. Основні поняття та властивості об'єктно-орієнтованого програмування (ООП). Класи C++

Концепція об'єктно-орієнтованого програмування, її відмінність від концепції структурного програмування. Компоненти об'єктно орієнтованої парадигми: об'єкт, повідомлення, клас, властивість, метод. Основні властивості ООП: абстракція, інкапсуляція, наслідування, поліморфізм. Інтерпретація основних понять ООП в термінах C++. Клас як розширення структурного типу. Синтаксис описання класу. Ідентифікатори доступу private, protected, public. Інкапсуляція даних та функцій. Призначення та створення конструкторів класу. Параметри конструктора. Призначення та створення деструктора класу. Приклади опису класів. Об'єкт як екземпляр деякого класу. Створення об'єкта. Доступ до властивостей та методів об'єкта. Використання вказівника this для одержання доступу до поточного об'єкта. Опис класу в двох файлах: інтерфейс класу (файл *.h), визначення функцій класу (файл *.cpp).

Тема 2.2. Структура та ієрархія класів, наслідування, поліморфізм

Поняття простого наслідування, базового класу, похідного класу, ієрархії класів. Синтаксис визначення похідного класу на основі базового. Відкриті та закриті похідні класи. Правила доступу до полів та функцій базових класів. Правила доступу до похідних класів та їх об'єктів. Створення ієрархії класів для простого наслідування. Приклади простого наслідування. Поняття поліморфізму та інтерпретація його в термінах віртуальних функцій C++. Зв'язування методів з об'єктами на етапі виконання програми – "пізні зв'язування". Зв'язування методів з об'єктами на етапі компіляції – "ранні зв'язування". Відмінності між двома способами зв'язування. Поняття поліморфного кластера. Реалізація віртуальних функцій. Приклади віртуальних функцій. Поняття множинного наслідування. Подання множинного наслідування у вигляді графів. Приклади множинного наслідування. Конфлікти, що можуть виникати під час множинного наслідування. Розв'язання конфліктів.

Тема 2.3. Перевантаження операторів у C++. Шаблони та шаблонні функції.

Способи перевантаження операцій у C++. Обмеження, що виникають під час перевантаження операцій. Використання класів для перевантаження операцій. Приклад класу для перевантаження операцій " + ", " * ". Перевантаження операцій new та delete. Перевантаження оператора присвоєння. Шаблони та шаблонні функції.

Тема 2.4. Організація абстрактних структур даних

Поняття лінійного списку. Методи організації та збереження списків. Використання масивів для збереження елементів списків. Спосіб зв'язаного збереження списків з використанням рекурсивно визначених структур. Огляд операцій, що виникають під час обробки списків. Реалізація операцій для послідовного збереження списків. Реалізація операцій для зв'язаного збереження списків. Двозв'язані списки. Реалізація основних операцій для двозв'язаних списків. Циклічні списки. Стек (структура LIFO - Last In First Out) як спеціальний тип лінійного списку. Поняття вершини стека. Організація та збереження елементів стека. Основні операції для роботи зі стеками. Черга (структура FIFO - First In First Out) як спеціальний тип лінійного списку. Поняття початку та кінця черги. Основні операції для роботи з чергами. Основні поняття та визначення: дерево, корінь, вузол, листок, дочірній вузол, рівень вузла, праве та ліве під дерево, глибина дерева. Способи збереження дерев. Алгоритми для проходження дерева: прямий, симетричний, обернений. Основні операції для роботи з деревами.

Тема 2.5. Управління компіляцією. Стандартна бібліотека STL

Призначення C++ препроцесора. Поняття та використання макророзстановки. Умовна компіляція. Основні компоненти STL. STL-рядки. Вектори. Ітератори. Алгоритми.

Перелік навчальних робіт з дисципліни «Основи теорії інформаційних систем»

Види робіт	К-сть балів
Лабораторне заняття №1. Тема: «Розробка програми з найпростішою структурою».	3
Лабораторне заняття №2. Тема: «Використання керуючих конструкцій. Умовний оператор».	3
Лабораторне заняття №3. Тема: «Використання керуючих конструкцій. Оператори циклу».	3
Лабораторне заняття №4. Тема: «Розробка програм по обробці масивів».	3
Лабораторне заняття №5. Тема: «Вказівники та адресна арифметика».	3
Лабораторне заняття №6. Тема: «Робота з динамічними масивами».	3
Лабораторне заняття №7. Тема: «Використання функцій, створених користувачем».	3
Лабораторне заняття №8. Тема: «Робота з масивами та функціями».	3
Лабораторне заняття №9. Тема: «Складені типи даних: структури, об'єднання, перерахування».	3
Лабораторне заняття №10. Тема: «Розробка програм з використанням найпростіших класів».	3
Лабораторне заняття №11. Тема: «Використання спеціальних методів при розробці класу».	3
Лабораторне заняття №12. Тема: «Принцип наслідування в програмах на принципі ООП».	3
Лабораторне заняття №13. Тема: «Принцип поліморфізму в програмах на принципі ООП».	3
Лабораторне заняття №14. Тема: «Організація виведення у вікно документа».	3
Лабораторне заняття №15. Тема: «Використання списків та стеків».	3
Лабораторне заняття №16. Тема: «Використання дерев».	3
Лабораторне заняття №17. Тема: «Використання бібліотеки STL. Умовна компіляція».	3
Модульний контроль	19
Разом: Аудиторна робота	70
Самостійна робота (СР)	30
Всього:	100

КОНТРОЛЬ ТА КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

При вивченні дисципліни використовуються наступні форми контролю знань студентів: поточний; модульний; підсумковий.

Поточний контроль передбачає перевірку теоретичних питань, самостійної роботи, практичних робіт та усне опитування по кожній практичній роботі. По даному виду контролю оцінювання знань здійснюється у відповідності до бального розподілу наведеного в попередній таблиці.

Модульний контроль передбачає виконання модульної контрольної роботи. Всі завдання оцінюються в 19 балів. Перше завдання (теоретичне) – 5 балів, друге завдання (практичне) – 7 балів, третє завдання (практичне) – 7 балів.

Формою підсумкового контролю є екзамен. Екзаменаційна оцінка (100 балів) є результатом виконання одного теоретичного питання (40 балів) та двох практичних завдань (2x30=60 балів).

Результуюча оцінка з дисципліни визначається як середня від балів набраних протягом семестру та отриманих на іспиті.

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний:

1. Рудий Т. В., Паранчук Я. С., Сенік В. В. Алгоритмізація та програмування. Частина 1. Структурне програмування : навчальний посібник. Львів : Львівський державний університет внутрішніх справ, 2022. 240 с.
2. Ковалюк Т. В. Алгоритмізація та програмування: підручник для студентів вищих навчальних закладів, що навчаються за напрямками "Комп'ютерні науки", "Комп'ютерна інженерія", "Програмна інженерія". Міністерство освіти і науки, молоді та спорту України. Львів :Видавництво "Магнолія 2006". 2019.
3. Grimes R. Beginning C++ Programming. Packt Publishing Ltd. 2017. 516 p.
4. Halterman R.L. Fundamentals of Programming C++. Southern Adventist University, 2018. 766 p.