



Структури даних і алгоритми

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>F Інформаційні технології</i>
Спеціальність	<i>F7 Комп'ютерна інженерія</i>
Освітня програма	<i>Комп'ютерні системи та мережі</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна</i>
Рік підготовки, семестр	<i>1 рік (1 семестр)</i>
Обсяг дисципліни	<i>5 кредитів (ECTS). Виділення часу -150 годин, включаючи лекції 30 год., лабораторні роботи 30 год, СРС 90 год.</i>
Семестровий контроль/ контрольні заходи	<i>1 семестр – екзамен</i>
Розклад занять	<i>Згідно розкладу на осінній семестр поточного навчального року за адресою http://rozklad.kpi.ua</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор д.т.н, професор Анатолій Михайлович Сергієнко , email Sergiyenko.Anatoliy@iit.kpi.ua Викладач з лабораторних занять магістр науковий: Русінов Володимир Володимирович, Email: VRusinovIO51@office365.fiot.kpi.ua</i>
Розміщення курсу	<i>На платформі дистанційного навчання «Сікорський»: https://classroom.google.com/c/NDIyNTg2NDY3NTY4?cjc=whbtqcx</i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Викладання алгоритмів та структур даних в університеті має такі аспекти:

Академічний аспект. Алгоритми та структури даних - це наука, яка дає фундаментальні знання. Мета - ознайомити студентів з основними поняттями та методами представлення даних та виконання з ними алгоритмів, вимірюючи складність алгоритму.

Професійний аспект. Знання структур даних та алгоритмів є обов'язковими для інженерів та програмістів, які розробляють програмне забезпечення для комп'ютера. Мета цього курсу – сформувати навички використання мови C для програмування базових алгоритмів, які використовуються у більшості програмного забезпечення універсальних та спеціалізованих систем.

Інтелектуальні та освітні аспекти. Вивчення структур даних та алгоритмів сприяє розвитку когнітивних навичок, що є суттєвим інтелектуальним фактором у процесі створення програмного забезпечення.

В результаті вивчення цієї дисципліни досягаються такі результати навчання.

Компетенції:

ЗК 01 Здатність до абстрактного мислення, аналізу та синтезу.

ЗК 02 Здатність вчитися і оволодівати сучасними знаннями.

ЗК 03 Здатність застосовувати знання у практичних ситуаціях.

ЗК 07 Вміння виявляти, ставити та вирішувати проблеми.

ФК 02 Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК 03 Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК 05 Здатність використовувати засоби і системи автоматизації проектування до розроблення компонентів комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем тощо.

ФК 11 Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

Прорамні результати навчання:

ПРН 01 Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.

ПРН 02 Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах.

ПРН 08 Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

ПРН 15 Вміти виконувати експериментальні дослідження за професійною тематикою.

ПРН 16 Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

ПРН 21 Якісно виконувати роботу з дотриманням вимог професійної етики.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Передумови, тобто дисципліни, вивчення яких має передувати вивченню цієї дисципліни:

- базовий рівень володіння англійською мовою не нижче А1.
- математична підготовка рівня середньої школи.

Знання та навички, отримані під час вивчення дисципліни «Алгоритми та структури даних», можуть бути застосовані під час вирішення навчальних та лабораторних завдань, що виникають під час засвоєння дисциплін:

- програмування;
- архітектура комп'ютера;
- алгоритми та методи обчислень;
- інженерія програмного забезпечення;
- організація баз даних;
- комп'ютерні мережі;
- паралельні та розподілені обчислення;
- комп'ютерні системи.

3. Зміст навчальної дисципліни

Тема 1: Основи мови C.

Предмет і мета предмета «структури даних та алгоритми». Поняття алгоритму. Архітектура з точки зору програміста. Історія мови C. Модель мови C. Об'єкти мови C. Операції та оператори мови C. Перетворення типів у виразах.

Тема 2: Алгоритми та програми.

Конструкції для проектування алгоритму. Блокові конструкції. Конструкції управління: конструкції розгалуження, повторення, перехід. Подання алгоритмів та оператори. Діаграми діяльності. Алгоритмічні блок-схеми. Основи структурного програмування.

Основи оцінки алгоритму. Часова складність. Просторова складність. Оцінка порядку складності. Ітеративні алгоритми. Типи алгоритмів: поліноміальні комплексні та NP-повні алгоритми. Вимірювання складності в програмі. Оптимізація алгоритму. Найпростіші алгоритмічні прийоми. Метод динамічного програмування.

Підпрограми. Функції та процедури. Контекст підпрограми, інтерфейс та реалізація. Реалізація підпрограми в архітектурі. Методи передачі параметрів. Структура програми C. Програмування функцій у C. Методи передачі параметрів у підпрограму.

Тема 3: Структури даних.

Концепція абстрактного типу даних. Функції над структурами даних. Класифікації структури даних. Статичні та динамічні структури даних. Скалярні структури даних. Комбіновані структури даних. Рівномірні та неоднорідні структури даних. Масиви.

Алгоритми зсуву масиву. Одно- та багатовимірне оголошення та зберігання масиву. Програмування дисплея консолі. Алгоритми переміщення 2-D масиву. Особливості роботи з квадратними матрицями.

Тема 4: Алгоритми з масивами.

Алгоритми пошуку. Алгоритм лінійного пошуку та його програмування. Алгоритми сортування та їх використання. Оцінка алгоритмів сортування. Сортування шляхом прямої вставки. Алгоритми сортування шляхом прямого вибору. Алгоритм швидкого сортування. Сортування злиттям. Сортування з підрахунком. Апаратна сортування. Алгоритми з рядками. Функції зі рядками. Структури. Множини. Юніони.

Тема 5: Рекурсивні алгоритми та їх програмування.

Рекурсивні алгоритми. Виклик рекурсивної функції та повернення з неї. Сигналізуюча функція. Порівняння рекурсивних та ітераційних алгоритмів.

4. Навчальні матеріали

4.1. Базова література

1. Керниган Б., Рітчі Д. Мова програмування Cі. 229 с.
2. Кормен Т. Г., Лейзерсон Ч. Е., Рівест Р. Л., Стайн К. Вступ до алгоритмів. Київ: К. І. С., 2019. 1288 с.
3. Макконелл. Дж. Основи сучасних алгоритмів. 2004. 368 с.

4.2. Додаткова література

4. Кормен Т. Г. Алгоритми доступно. Київ: К.І.С, 2021. 206 с.
5. Ковалюк Т. В. Основи програмування. Київ: BHV. 2005. 384 с.
6. Непийвода Н. Н., Скопін І. Н.. Основи програмування. 2002. 919 с.

7. Вірт, Алгоритми та структури даних Н.. 358 с. ДМК. 2022. 272 с.

8. Sergiyenko A., Serhienko A., Romankevich V. "Genetic Programming of Pipelined Datapaths for FPGA," 2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO), Kyiv, Ukraine, 2020, pp. 802-806, doi: 10.1109/ELNANO50318.2020.9088773.

4.3. Інформаційні ресурси

На платформі дистанційного навчання «Сікорський»:
<https://classroom.google.com/c/NDIyNTg2NDY3NTY4?cjc=whbtqcx>

Курс відеолекцій – на платформі дистанційного навчання Big Blue Button:
<https://bbb.comsys.kpi.ua/b/ana-fxo-y2d-ykw>

Слайди лекцій та керівництва до лабораторних робіт <https://kanyevsky.kpi.ua/en/for-students/>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальний зміст дисципліни складається з лекцій та лабораторних робіт.

Лекції.

Тема 1: Основи мови С.

Лекція 1. Вступ.

Предмет і мета курсу «структур даних та алгоритмів». Сучасний рівень обчислювальної техніки. Навчальні матеріали. Програмне забезпечення для використання. Система оцінок. Історія алгоритмів та розрахунків. Концепція алгоритму. Обчислювальний процес. Граф алгоритму. Архітектура комп'ютера та його історія. Ієрархія архітектури. Архітектура з точки зору програміста.

Лекція 2. Модель мови С.

Модель фон Неймана. Історія мови С. Стандарти мови С. Мова С для комп'ютера PDP-11. Модель мови С. Об'єкти мови С: коментарі, символи, ключові слова, ідентифікатори, літерали. Дані та їх види. Змінні. Локальні та глобальні об'єкти.

Лекція 3. Операції з мовою С

Операції та оператори. Арифметичні та логічні операції. Операції призначення. Пріоритети операцій. Перетворення типів у виразах.

Тема 2: Алгоритми та програми.

Лекція 4. Конструкції для проектування алгоритму

Блокові конструкції. Конструкції управління: конструкції розгалуження, повторення, перехід. Умовні, безумовні та селективні конструкції. Однорівневі конструкції переходів.

Лекція 5. Подання алгоритмів та оператори.

Діаграми діяльності. Блок-схеми алгоритмів. Оператори умовного та безумовного переходу. Функції вводу-виводу. Теорія схем програм. Основи структурного програмування.

Лекція 6. Основи оцінки алгоритмів.

Часова складність. Просторова складність. Стандартні моделі для оцінки часової складності. Розрахунок часової складності в програмах С. Оцінка порядку складності. Ітеративні алгоритми. Приклад алгоритму та розрахунок його складності.

Лекція 7. Основи оцінки алгоритму (продовження)

Спрощена оцінка складності. Типи алгоритмів: поліноміальна складність та NP-повні алгоритми. Вимірювання складності в програмі. Приклад алгоритму та розрахунок його складності.

Лекція 8. Оптимізація алгоритму.

Найпростіші алгоритмічні прийоми. Використання множника зі змінним знаком. Перевірка парних номерів. Використання прапорця знака. Розгортання циклів. Заміна множення на додавання. Метод динамічного програмування. Мінімізація умовних гілок.

Лекція 9. Підпрограми.

Функції та процедури. Контекст, інтерфейс та реалізація підпрограми. Виклик підпрограми. Реалізація підпрограми в архітектурі. Методи передачі параметрів. Функції на мові C. Структура програми C. Програмування функцій у C. Передача параметрів за значенням. Параметр, що передається за посиланням. Параметр типу const. Ключове слово static. Конвенція про виклик підпрограм у мові C.

Тема 3: Структури даних.

Лекція 10. Структури даних.

Структури даних. Концепція абстрактного типу даних. Функції над структурами даних. Класифікація структур даних. Статичні та динамічні структури даних. Скалярні структури даних. Комбіновані структури даних. Рівномірні та неоднорідні структури даних. Масиви. Одновимірний масив, його зберігання та доступ.

Лекція 11. Масиви та їх зміна.

Алгоритми зсуву масиву. Циклічний зсув масиву в одну позицію. Циклічний зсув масиву на k позицій. Обмін блоками. Циклічний зсув, що зберігає середню частину масиву. Циклічний зсув із переворотами.

Лекція 12. Двовимірні масиви

Оголошення та зберігання багатовимірних масиву. Ініціалізація двовимірних масиву. Програмування дисплея консолі. Бібліотека Windows та її функції для програмування прямого доступу до буфера консолі. Алгоритми обходу двовимірних масиву. Особливості роботи з квадратними матрицями.

Тема 4: Алгоритми з масивами.

Лекція 13. Алгоритми пошуку.

Задача пошуку. Алгоритм лінійного пошуку та його програмування. Класичний алгоритм лінійного пошуку. Лінійний пошук з бар'єром. Двійковий пошук та його дві варіації. Прямий пошук рядків. КМП-пошук. Алгоритм Рабіна-Карпа. Алгоритми пошуку з хеш-функцією.

Лекція 14. Алгоритми сортування.

Алгоритми сортування та їх використання. Оцінка алгоритмів сортування. Сортування прямою вставкою. Алгоритм з лінійним пошуком за вставленим елементом. Алгоритм з лінійним пошуком від елемента, який вставляється. Алгоритм з лінійним пошуком від елемента, що вставляється з бар'єром. Алгоритм з двійковим пошуком. Аналіз алгоритмів прямої вставки.

Лекція 15. Алгоритми сортування (продовження)

Алгоритми сортування шляхом прямого вибору. Алгоритм прямого обміну. Алгоритм бульбашкового сортування. Алгоритм шейкерного сортування. Алгоритм Шела.

Лекція 16. Алгоритми сортування (закінчення)

Алгоритм швидкої сортування. Сортування злиттям. Сортування з підрахунком. Апаратне сортування. Мережа бітонного сортування. Мережа бульбашкового сортування. Мережа сортування парно-непарним злиттям.

Лекція 17. Алгоритми з рядками

Рядки. Представлення рядків в комп'ютерах. Функції зі рядками. Структури. Множини. Юніти.

Тема 5: Рекурсивні алгоритми та їх програмування.

Лекція 18.

Рекурсивні алгоритми. Виклик рекурсивної функції та повернення з неї. Сигналізуюча функція. Порівняння рекурсивних та ітераційних алгоритмів.

Деталізація тем лекційних занять, конспект лекцій та відеоматеріали розміщено на платформі дистанційного навчання «Сікорській»

<https://classroom.google.com/c/NDIyNTg2NDY3NTY4?cjc=whbtqcx>

Лабораторні роботи

Мета лабораторних робіт – експериментальне дослідження та випробування алгоритмів та структур даних, що вивчаються, при програмуванні завдань лабораторних робіт.

Тематика лабораторних робіт:

Лабораторна робота 1.

Розгалужені алгоритми.

Лабораторна робота 2.

Циклічні алгоритми та динамічне програмування.

Лабораторна робота 3.

Алгоритми обходу двовимірних матриць.

Лабораторна робота 4.

Двійкові алгоритми пошуку.

Лабораторна робота 5.

Алгоритми сортування.

Загальні методичні рекомендації для дистанційного виконання:

1. Зосередитись на програмуванні. Основний навчальний чинник курсу — експериментальне випробування алгоритмів та структур даних, що вивчаються, при програмуванні завдань лабораторних робіт на власному комп'ютері студента. При цьому програмування можна виконувати у будь-яких сприятливих умовах у довільний час. При відсутності центрального електропостачання, програмування можна виконувати використовуючи заряд акумулятора.

2. Використання матеріалів, які доступні в мережі Інтернет. При неможливості бути присутнім на лекції кожен студент може продивитись у запису, який зберігається на сервері кафедри, у довільний час, який є зручним для студента, навіть з повтором, якщо щось є незрозумілим з першого разу. Також в Інтернеті є багато довідкового матеріалу, що стосується курсу, який можна знайти за пошукачем чи у Вікіпедії або користуючись генеративним штучним інтелектом.

3. Комунікація та підтримка. Використовувати онлайн-інструменти для зворотного зв'язку, обговорення завдань та підтримки студентів у процесі виконання проєктів. Важливо мати регулярні консультації та можливість допомогти студентам у разі виникнення питань.

6. Самостійна робота здобувача вищої освіти

Види самостійної роботи для студентів денної форми навчання (90 годин):

- підготовка до лекцій через вивчення попереднього лекційного матеріалу, а також літературних джерел, на яких ґрунтується матеріал попередніх лекцій (1 година x 15 лекцій = 15 годин);
- підготовка до лабораторної роботи через ознайомлення із завданням та методичними вказівками до лабораторних робіт, включаючи вивчення теоретичного матеріалу, необхідного для відповіді на контрольні запитання до лабораторних робіт, складання і програмування алгоритму, компіляцію та відлагодження програми, розгортання програмного середовища та завантаження вихідних кодів програмного забезпечення для виконання лабораторних робіт, робота з системою контролю версій Git, а також оформлення звіту. Кожна лабораторна робота вимагає для підготовки та виконання двох тижнів і витрат до 12 годин часу (5 x 12 годин = 60 годин);
- підготовка до екзамену (15 годин);

Теми на самостійне опрацювання

Тема 1: Основи мови C.

Основи роботи з IDE. Введення в систему керування версіями Git. (5 годин).

Тема 2: Алгоритми та програми.

Модель машини Тюрінга і основи оцінки алгоритму. Підпрограми з динамічним зв'язуванням. Взаємодія програми з операційною системою. Передача адрес через дескриптори. (5 годин).

Тема 3: Структури даних.

Файлові структури даних. Функції роботи з файлами. (5 годин).

Тема 4: Алгоритми з масивами.

Алгоритми з рядками. Рядки. Представлення рядків в комп'ютерах. Функції з рядками. Записи, структури. Множини. Юніти. (5 годин).

Матеріали для самостійного опрацювання, теми, питання для самоконтролю та контрольні запитання знаходяться на платформі дистанційного навчання «Сікорський» (Classroom).
<https://classroom.google.com/c/NDIyNTg2NDY3NTY4?cjc=whbtqcx>

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог до студентів:

- студент зобов'язаний відвідувати лекції та лабораторні заняття, активно працювати над засвоєнням викладеного на них матеріалу;
- на лекції лектор використовує власний презентаційний матеріал;
- лабораторні роботи захищаються у два етапи: перший етап - студенти виконують завдання, складають електронний звіт та надсилають викладачу; другий етап - захист лабораторних робіт у лабораторії. Контроль знань на лабораторних роботах здійснюється шляхом перевірки звіту про лабораторні роботи, а також питань з матеріалів робіт.

Дедлайни. Для виконання лабораторних робіт встановлюються дедлайни. Дедлайни уточнюються і оприлюднюються викладачем на початку семестру.

Заохочувальні бали (бонуси) виставляються за: активну участь на лекціях та лабораторних заняттях, ведення конспекту, підготовку повідомлення з презентацією по одній із тем СРС дисципліни тощо. Кількість заохочуваних балів не більше 5.

Лабораторні роботи. Виконанню кожної лабораторної роботи передуює виконання індивідуального завдання і оформлення його у вигляді протоколу. Протокол виконання лабораторного завдання має бути вислано на пошту викладача не пізніше ніж за добу до лабораторного завдання. До протоколу додаються посилання на Git з кодом програм (за необхідності) і посилання на всі необхідні супровідні матеріали для захисту, зокрема на відеозапис виконання лабораторного завдання (за необхідності, в дистанційному режимі або за використання віддаленого устаткування, тощо). Студент, який відвідав заняття без підготовленого протоколу до захисту лабораторної роботи не допускається. Першим етапом студент захищає результати виконання індивідуального завдання до лабораторної роботи, на другому етапі – захищає теорію шляхом усного опитування або тестування. В дистанційному режимі більшість лабораторних робіт супроводжуються тестами для оцінки вивченого теоретичного та лабораторного матеріалу замість усного опитування. Бали отримані за виконання лабораторної роботи - за тест та за протокол складають оцінку за лабораторну роботу. Тестування проводиться на лабораторному занятті після перевірки результатів виконання індивідуальної роботи. Студент, який не виконав індивідуальне завдання до лабораторної роботи до тесту не допускається.

Виконання лабораторних робіт є обов'язковими для допуску до семестрового контролю (екзамену). Загальною умовою допуску до семестрового контролю (екзамену) є зарахування всіх лабораторних робіт та загальний рейтинг не менше 60 балів.

Оцінка, яку студент може отримати за виконання кожної лабораторної роботи наведені у розділі 8 силабуса.

Здобувачі, які виконали всі умови допуску до екзамену та мають рейтингову оцінку менше 60 балів, а також здобувачі, які бажають підвищити свою рейтингову оцінку, на останньому за розкладом занятті виконують контрольну роботу та співбесіду.

До індивідуальних семестрових завдань вноситься індивідуальна робота студента що стосується виконання лабораторних робіт. Кількість балів за індивідуальну роботу за семестр дорівнює від 38 до 60 балів.

Правила відвідування занять. Відвідування занять є вільним, бали за присутність на лекціях та лабораторних заняттях не додаються, відповідно штрафні бали не передбачаються. Втім, вагома частина рейтингу формується через активну участь у заходах на лабораторних заняттях.

Дистанційний режим навчання. У разі запровадження обмежень на відвідування університету, пов'язаних з введенням карантину або режиму воєнного стану в державі, освітній процес здійснюється у дистанційному режимі відповідно до Положення про дистанційне навчання в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/index.php/node/188>), Регламенту організації освітнього процесу в дистанційному режимі (<https://profkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distantsiynomu-rezhimi>) та Регламенту проведення семестрового контролю в дистанційному режимі (<https://osvita.kpi.ua/node/148>). У режимі дистанційного навчання заняття відбуваються у вигляді онлайн-конференції на платформах BigBlueButton, Google Meet, Zoom. Посилання на конференцію видається на початку семестру і розміщується в АС «Електронний кампус». З метою забезпечення якісної підготовки здобувачів, дистанційний курс дисципліни розміщено на Платформа дистанційного навчання «Сікорський» (<https://www.sikorsky-distance.org>). Результати оцінювання висвітлюють у АС «Електронний кампус» на особистій сторінці здобувача (<https://ecampus.kpi.ua>).

Правила поведінки на заняттях. На заняттях слід дотримуватись норм етичної поведінки визначених у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>). На території університету

здобувачі мають поводити себе відповідно до Правил внутрішнього розпорядку (<https://kpi.ua/admin-rule>).

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Положення про визнання результатів навчання, набутих у неформальній / інформальній освіті (<https://osvita.kpi.ua/index.php/node/179>) регламентує визнання результатів навчання, набутих у неформальній/інформальній освіті.

Неформальна освіта. У разі проходження здобувачем онлайн курсу чи іншого елементу неформальної освіти здобувачеві/здобувачці можуть бути зараховані окремі змістовні модулі або теми дисципліни. В такому разі здобувач звільняється від виконання відповідних лабораторних робіт, отримуючи за них максимальний бал відповідно до рейтингової системи оцінювання. У разі навчання на неформальній освіті за самостійного її обрання проходиться процедура валідації, що передбачає подання здобувачем заяви на ім'я декана, декларації підтверджувальних документів. Рішення про визнання чи не визнання приймається комісією у складі завідувача кафедри, викладача, гаранта освітньо-професійної програми.

Політика використання штучного інтелекту. Використання штучного інтелекту регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/1225>). Усі завдання, як під час виконання навчальних завдань з дисципліни, так і індивідуальні завдання, мають бути результатом власної оригінальної роботи здобувача. Використання ШІ для автоматичної генерації відповідей без подальшого їх аналізу та доопрацювання заборонено.

Порушення академічної доброчесності: використання плагіату. У разі виявлення фактів використання плагіату під час виконання навчальних завдань, студент втрачає право на подальше виконання поточних завдань у межах навчальної дисципліни до з'ясування обставин порушення. Студент зобов'язаний аргументовано пояснити причини використання плагіату, а також засвідчити готовність дотримуватись принципів академічної доброчесності в подальшому навчанні; повторно виконати завдання або, за потреби, виконати додаткове завдання відповідно до бачення викладача з урахуванням вимог щодо самостійності.

Зазначені дії регламентуються Кодексом честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>)

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

На першому занятті студенти знайомляться з політикою оцінювання, яка ґрунтується на Положенні про систему оцінювання результатів навчання https://document.kpi.ua/files/2020_1-273.pdf. Рейтинг студента на курсі складається з балів, які він/вона отримує за захищені лабораторні роботи (R1) та екзаменаційну оцінку (R2).

$$R_s = R_1 + R_2 = 100 \text{ балів}$$

В результаті максимальний середній бал дорівнює:

$$6 \text{ лабораторних робіт} \times 10 \text{ балів} = 60 \text{ балів}$$

$$\text{екзамен} = 40 \text{ балів.}$$

Рейтинг вважається дійсним за умови, що студент виконав усі і захистив успішно принаймні 4 лабораторних роботи.

Згідно з університетськими правилами щодо моніторингу успішності студента (https://kpi.ua/document_control), існує два тижні оцінювання (атестація), зазвичай протягом 7/8 та 14/15 тижнів семестру, коли оцінюється прогрес у навчанні студентів та відповідність до критеріїв політики оцінювання курсу.

Умовою першої атестації є отримання не менше 10 балів (на момент атестації). Умова другої атестації - набрати не менше 30 балів (на момент атестації).

Критеріями оцінки є:

- Виконання лабораторних робіт:

- бездоганна робота оцінюється у 11 балів, правильні відповіді на усі питання при захисті;
- є певні недоліки в оформленій роботі - 10-9 балів, є неточні відповіді на питання;
- є деякі недоліки у виконанні робочої програми - 8-6 балів, окремі відповіді на питання є неправильними;

Робота не виконана або не захищена - 0 балів.

За роботу, подану вчасно, діє заохочення - 1 бал (сумарно не більше - 5 балів).

- Результати екзамену оцінюються 40 балами. Екзаменаційна робота складається з 25 тестових питань, а також практичного завдання (скласти програму) зі списку, наведеного в додатку до робочої програми.

За кожну правильну відповідь на тестове питання нараховується 1 бал. Відповідь на практичне завдання оцінюється 5 балами за такими критеріями:

- "відмінно" - правильний текст програми з коментарями - 5 балів;
- "добре" - текст програми, загалом правильний, але не використаний прийом оптимізації - 4-3 балів;
- "задовільно" - є деякі принципові помилки в тексті програми - 2 - 1 бал;
- "незадовільно" - незадовільна відповідь - 0 балів.

Студенти можуть отримати до 6 заохочувальних балів за виконання творчих робіт з кредитного модуля (складання тез, участь у конкурсах, у дослідженнях тощо).

Таблиця. Остаточна оцінка RS приймається за системою оцінювання університету

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація про курс

Питання в модульній контрольній роботі виглядають як наступні.

1. Обчислювальний алгоритм – це а) послідовність дій, виконання якої приводить до очікуваного результату, б) обчислювальний процес, що заданий на чітко описаній моделі обчислювача, в) граф або програма, обхід частин яких відповідає деякому обчислювальному процесу.
2. В ЕОМ використовуються 6-бітові літери. Алфавіт цієї ЕОМ вміщує: а) 32 цифри і знаки, б) 55 цифр, букв і знаків, в) 65 цифр, букв і знаків.
3. Що означає у мові Сі запис: \0 — а) ділення на нуль без переривання, б), символ NULL в) ознака кінця рядка?
4. Змінна типу int займає в машині об'єм — а) 16 біт, б) 32 біт, в) 16 або 32 біт .
5. Вираз: a&&b/c & d; еквівалентний виразу — а) (a&&(b/c)) & d;, б) a&&((b/c) & d);, в) (a&&b)/(c&d);.
6. Результатом операції 12 || 6 є — а) 6, б) 1, в) 14 .

7. Результатом операції `switch (3 >> 1){case 0: a = 1; case 1: a = 2; break; default: a=3;}` є: а) 1, б) 2, в) 3.
8. Блок — це: а) виділена частина програми, змінні якої мають локальну дію, б) виділена частина програми, яка має інтерфейс для вхідних і вихідних змінних, в) частина структури комп'ютера.
9. цикл з передумовою — це цикл, який а) перестає виконуватись за умовою, б) виконується за умовою, в) це цикл, який починає виконуватись за умовою.
10. Щоб цикл з лічильником не зациквився, слід забезпечити — а) умови обов'язкового виходу з циклу, б) щоб справджувалась умова оператора `for` хоча б в одній з ітерацій, в) щоб змінна лічильника була цілого типу.
11. Оператор `continue` призначений для — а) продовження роботи програмного модуля після переривання, б) виходу з циклу, в) переходу на наступну ітерацію циклу.
12. Оператор `for (i = 1; ;i++){...}` — виконує цикл, — а) оператор є неправильним, б) якщо у його блоці є оператор виходу з циклу, в) якщо у його блоці є оператор виходу з циклу, інакше — нескінченний цикл.
13. При виконанні фрагменту: `int A[3] = {1,2,3}; printf("%d",A[3]);` буде виводитись — а) 3, б) 0, в) невідоме число.
14. Дані статичної структури — це дані, — а) які складаються з констант, б) взаєморозташування та взаємозв'язки яких є постійними, в) які мають незмінне розташування під час виконання програми.
15. У програмі на Сі масив задано як `int a[2][3] = { 1, 2, 3, 4, 5, 6 }`; Чому дорівнює елемент `a[2][3]` а) NULL, б) невідоме число, в) 6 ?.
16. Сортування з підрахунком ґрунтується на тому, що підраховується — а) кількість елементів у масиві однакової величини, б) дистанція, на яку слід перемістити черговий елемент на черговому кроці впорядкування масиву, в) довжина частини масиву, яка потребує сортування.
17. `a` і `b` — цілі числа. Чи можлива рівність `a[b] = b[a]` ? а) так, бо `a[b] = *(a + b) = b[a]`, б) ні, бо в `a[b]` `b`-це індекс, а `a`-ім'я масиву, а в `b[a]`- навпаки, в) ні, бо `a` і `b` різнотипні ідентифікатори.
18. Що виконує вираз `*(a + i) + j` ? а) виконує доступ до комірки `a[i][j]`, б) посилання на суму `a+i`, до якого додається `j` і робиться наступне посилання, в) множення `a+i` на нуль і повертання вказівника на `j`.
19. Яка відміна між `structure` та `union`? а) в `union` поля перекриваються і не можуть використовуватись одночасно як в `structure`, б) `structure` зберігає кілька даних, а `union` — одне, в) в `structure` можна читати одне поле і записувати в інше, а в `union` — не можна.
20. Що виконає послідовність операторів: `enum set{a,b,c} s=b; printf ("%d ",s);` ? а) надрукує 1, б) надрукує 0, в) надрукує `b`.
21. Оператор `char ch = 'K';` записує в комірку `ch` а) ASCII-код літери `K` б) символ `K` в) символ `K`, за яким слідує символ `NULL`.
22. Є присвоювання `m.e.y = 2021;` Що є правильним? а) структура `e` вкладена в структуру `m` б) структура `y` вкладена в структуру `e` в) структура `m` вкладена в структуру `e`.
23. Рядок з завершальним нулем зберігається в пам'яті як а) послідовність символів, останній з якої є символ `NULL` б) масив символів, довжина рядка зберігається в першій комірці масиву, а в останній — нуль в) послідовність символів, останній з якої є число 0.
24. Задано перелічний тип `enum states{s1=2,s2,s3};` Що виконає послідовність операторів: `enum states S; S=s2; printf("%d",S);` ? а) надрукує 3 б) надрукує 1 в) надрукує `s2`.

25. Рекурсивний спуск – це а) послідовність викликів рекурсивної функції до точки першого повернення б) маршрут в графі при виконанні пошуку вглиб в) маршрут в бінарному дереві при пошуку за ключем.

II Розробити алгоритм і програму, що вирішує задачу. Є число з плаваючою комою X . Розрахувати масив $X[n]$, у якому $X[0] = X$; $X[i] = \sum_{k=1}^i (-X)^k (k+1)$; Число операцій має бути мінімізоване.

Робочу програму навчальної дисципліни (силабус):

Складено д.т.н., проф. А. М. Сергієнко

Ухвалено кафедрою обчислювальної техніки (протокол №18 від 24.06.2026)

Погоджено Методичною комісією ФІОТ (протокол №12 від 26.06.2026)