



АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ.

ЧАСТИНА 1. БАЗОВІ КОНЦЕПЦІЇ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>F Інформаційні технології</i>
Спеціальність	<i>F3 Комп'ютерні науки</i>
Освітня програма	<i>Цифрові технології в енергетиці</i>
Статус дисципліни	<i>Нормативна. Цикл професійної підготовки</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>1 курс, осінній семестр</i>
Обсяг дисципліни	<i>90 годин / 3 кредити ЄКТС (лекції – 30 год., лабораторні заняття – 14 год., СРС – 46 год.)</i>
Семестровий контроль/ контрольні заходи	<i>Залік, МКР</i>
Розклад занять	<i>http://roz.kpi.ua/</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викла- дачів	<i>Лекції і лабораторні заняття проводить: канд. техн .наук, доцент, Кублій Лариса Іванівна kublji_l_i@ukr.net, тел. 063-71-91-231 (+Телеграм), 097-558-27-17</i>
Розміщення курсу	<i>Кампус</i>

Програма навчальної дисципліни

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

Знання мов програмування, знання і застосування існуючих і розробка нових алгоритмів є невід'ємною складовою роботи кожного програміста. Згідно з робочим навчальним планом кредитний модуль "Алгоритмізація та програмування. Частина 1. Базові концепції програмування" (ПО-6.1) дисципліни "Алгоритмізація та програмування" викладається студентам першого року підготовки першого (бакалаврського) рівня вищої освіти ступеня "бакалавр" спеціальності "Комп'ютерні науки" освітньої програми "Цифрові технології в енергетиці" у першому навчальному семестрі. У рамках вивчення матеріалу модуля базовими є концепції структурного і модульного програмування. Студенти вивчають мову програмування C, знайомляться з системами числення, поданням числових даних у комп'ютері і основні алгоритми — класичні і сучасні, лінійні, з розгалуженнями, ітераційні і рекурсивні — зокрема, алгоритм Евкліда знаходження найбільшого спільного дільника, решето Ератосфена пошуку простих чисел, схему Ньютона для обчислення коренів, прості алгоритми сортування елементів масивів, алгоритми, які базуються на використанні побітових операцій, та багато інших, значну увагу приділено врахуванню можливості арифметичного переповнення при роботі з цілими числами і точності обчислень при роботі з дійсними числами.

Метою опанування дисципліни є формування у студентів загальних і фахових компетентностей:

- здатність до абстрактного мислення, аналізу та синтезу (ЗК 01);
- здатність застосовувати знання у практичних ситуаціях (ЗК 02);
- знання та розуміння предметної області та розуміння професійної діяльності (ЗК 03);
- здатність вчитися й оволодівати сучасними знаннями (ЗК 06);

— здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їхньої ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем (ФК 03);

— здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління (ФК 08);

— використання принципів структурного програмування, основних структур даних під час реалізації алгоритмів професійних завдань;

— володіння алгоритмічним мисленням, методами реалізації програмного забезпечення з урахуванням вимог до його якості, надійності, виробничих характеристик.

Предметом навчальної дисципліни є вивчення мови програмування C, простих структур даних і типових алгоритмів роботи з ними.

Студенти після засвоєння кредитного модуля мають продемонструвати такі результати навчання:

знання:

- основ організації обчислювального процесу на ЕОМ;
- систем числення;
- теоретичних основ алгоритмізації, проектування та тестування програм;
- методів структурного програмування;
- алгоритмів розв'язування типових задач;
- засобів програмування алгоритмічної мови C;

вміння:

- застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук (ПРН 01);
- 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 (ПРН 05);
- 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 (ПРН 09);
- застосовувати мови програмування (зокрема мову програмування C);
- проектувати компоненти програмного забезпечення;
- проектувати людино-машинний інтерфейс інформаційних систем;
- розробляти оптимальні алгоритми для широкого кола задач;
- реалізувати алгоритми мовою програмування C як закінченого програмного продукту;
- виконувати аналіз коректності програм, їхнє налагодження та тестування з використанням сучасних технологій програмування;
- застосовувати набуті базові знання з дисципліни в професійній діяльності під час розробки, налагодження та експлуатації ІС та технологій;
- застосовувати принципи структурного програмування при проектуванні і розробці програм;
- правильно вибирати й застосовувати основні структури даних під час програмної реалізації алгоритмів;
- аналізувати та програмувати виключні ситуації у програмних продуктах;
- застосовувати типові алгоритми обробки даних.

Додаткові знання:

- особливостей предметного середовища, застосування відповідних структур даних, спеціалізованих алгоритмів, різних мовних конструкцій, організації та використання різних структур даних.

Додаткові вміння:

- конструювати, кодувати, тестувати, налагоджувати та супроводжувати програмний код.

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

У структурно-логічній схемі навчання зазначена дисципліна розміщена у першому семестрі першого року навчання. Дисципліна “Алгоритмізація та програмування. Частина 1. Базові концепції програмування” не має дисциплін, які її забезпечують, вона вивчається на основі знань, отриманих у середній загальноосвітній школі. Компетенції, отримані студентами в процесі вивчення цього модуля, застосовуються при вивченні навчальних дисциплін: “Алгоритмізація та програмування. Частина 2. Процедурне програмування”, “Проектування та аналіз обчислювальних алгоритмів”, “Об’єктно-орієнтоване програмування”, “Технології розробки програмного забезпечення”.

3. Зміст навчальної дисципліни

Розділ 1. Організація програм

Тема 1.1. Архітектура комп’ютерів. Системи числення

Тема 1.2. Поняття алгоритму. Типові алгоритмічні структури

Тема 1.3. Елементи алгоритмічних мов: концепція типів даних, імена, значення, вказівники, змінні, константи, операції, вирази

Тема 1.4. Базові структури програмування: послідовність, розгалуження, цикли

Тема 1.5. Процедурно-орієнтоване програмування. Рекурсія

Розділ 2. Структури даних і алгоритми

Тема 2.1. Організація даних (масиви, рядки, структури) та алгоритми їхньої обробки

Тема 2.2. Алгоритмізація типових обчислювальних задач

4. Навчальні матеріали та ресурси

Базова література

1. Кублій Л.І. Алгоритми та структури даних. Основи алгоритмізації [Електронний ресурс]: підручник для здобувачів ступеня бакалавра за спеціальністю 121 “Інженерія програмного забезпечення” / Л.І. Кублій; КПІ ім. Ігоря Сікорського. — Електронні текстові дані (1 файл: 21,3 Мбайт). — Київ: КПІ ім. Ігоря Сікорського, 2022. — 528 с. — <https://ela.kpi.ua/handle/123456789/48282>.
2. Кублій Л. І. Алгоритмізація та програмування. Практикум [Електронний ресурс]: навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 122 “Комп’ютерні науки” / Л.І. Кублій; КПІ ім. Ігоря Сікорського. — Електронні текстові дані (1 файл: 28,15 Мбайт). — Київ: КПІ ім. Ігоря Сікорського, 2019. — 209 с. — ela.kpi.ua/handle/123456789/28216
3. Kernighan Brian W., Ritchie Dennis M. The C Programming Language: Second edition. — USA: Prentice Hall, 1988. — 288 p. — http://cslabcms.nju.edu.cn/problem_solving/images/c/cc/The_C_Programming_Language_%282nd_Edition_Ritchie_Kernighan%29.pdf (Керніган Б., Річі Д. Мова програмування С. Пер. з англ., 2-е вид. — 2013. — 232 с. — <http://programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>)
4. Шпак З.Я. Програмування мовою С. — Львів: Оріяна-Нова, 2006. — 432 с. — http://pdf.lib.vntu.edu.ua/books/2017/Shpak_2006_432.pdf

Додаткова література

5. Войтенко В.В., Морозов А.В. С/С++: Теорія та практика: навч.-метод. Посібник. 2-е вид. — Житомир: ЖДТУ, 2004. — 325 с. — <http://programming.in.ua/programming/c-language/183-c-or-c-book-voytenko-and-morozov.html>
6. Гарасимчук О. І. Генератори псевдовипадкових чисел, їх застосування, класифікація, основні методи побудови і оцінка якості [Електронний ресурс] / О. І. Гарасимчук, В. М. Максимович // Науково-технічний журнал «Захист інформації». — 2003. — № 30. — С. 29-36. — Режим доступу: <http://jrn1.nau.edu.ua/index.php/ZI/article/viewFile/4270/4405> (дата звернення: 12.02.2022).
7. Новотарський М.А. Алгоритми та методи обчислень [Електронний ресурс] : навч. посіб. для студ. спеціальностей 121 “Інженерія програмного забезпечення”, спеціалізації “Програмне забезпечення високопродуктивних комп’ютерних систем та мереж” та 123 “Комп’ютерна інженерія”, спеціалізації “Комп’ютерні системи та мережі”. — Електронні текстові дані (1 файл: 4648 Кбайт). — Київ : КПІ ім. Ігоря Сікорського, 2019. — 407 с. — https://ela.kpi.ua/bitstream/123456789/27864/1/Alhorytmy_ta_metody_obchislenn.pdf .
8. Fog Agner. 4. Instruction tables [Electronic resource]. — Technical University of Denmark, 22.03.2021. — 424 p. — https://www.agner.org/optimize/instruction_tables.pdf (date of access: 23.12.2021).
9. IEEE Standard for Floating-Point Arithmetic. — New York, 2008. — 58 p.

Інформаційні ресурси

1. Кампус НТУУ “КПІ” — <http://login.kpi.ua/>
2. Науково-технічна бібліотека НТУУ “КПІ імені Ігоря Сікорського” — <https://www.library.kpi.ua/>

5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальна дисципліна охоплює 30 годин лекцій і 14 годин лабораторних занять, передбачає виконання модульної контрольної роботи, яка проводиться на останньому лекційному занятті і триває 2 академічні години.

На лекціях студенти одержують основний теоретичний матеріал. Основні завдання циклу лабораторних занять полягають в тому, щоб студенти закріпили матеріал лекцій і отримали практичні навички у розробці алгоритмів і проектуванні програм з використанням сучасних технологій програмування.

За семестровим (кредитним) модулем передбачається одна модульна контрольна робота, яка проводиться в кінці семестру. Мета модульної контрольної роботи полягає в перевірці засвоєння матеріалу дисципліни. Контрольна робота містить 5 завдань. Для проведення контрольної роботи виділяється 2 години за рахунок лекційних занять. На контрольну роботу виносяться такі теми: програмування арифметичних виразів, програмування умовних операторів, обробка одновимірних і двовимірних масивів, алгоритми сортування.

Термін виконання (тиждень)	Назви розділів, тем, занять
Розділ 1. Організація програм	
Тема 1.1. Архітектура комп'ютерів. Системи числення	
1	<u>Лекція 1. Організація обчислювального процесу на комп'ютері. Системи числення</u> Коротка історична довідка. Розвиток кібернетики та обчислювальної техніки в Україні. Структура і принципи роботи персонального комп'ютера. Системи числення. Подання чисел у позиційних системах (двійкова, вісімкова, шіснадцяткова системи числення). Подання числових даних у комп'ютері: цілі й дійсні числа. Особливості роботи з дійсними числами. (стор. 61-112 [1], стор. 9-15 [2]; [8]).
1	<u>Лабораторна робота 1.</u> Основи роботи в інтегрованому середовищі Visual Studio (консольний режим C++). Оформлення звітів з лабораторних робіт засобами текстового редактора Word (побудова блок-схем, вставка формул, подання результатів роботи). Позиційні системи числення. Робота з дійсними числами на комп'ютері.
Тема 1.2. Поняття алгоритму. Типові алгоритмічні структури	
2	<u>Лекція 2. Алгоритми й алгоритмічні структури</u> Поняття алгоритму. Властивості алгоритмів. Складність алгоритмів. Способи подання алгоритмів. Базові алгоритмічні структури. Основні характеристики алгоритмічної мови С. Структура програми мовою С. (стор. 9-36 [1], стор. 15-18 [2], стор. 27-30 [5], стор. 19-40 [7]).
Тема 1.3. Елементи алгоритмічних мов: концепція типів даних, імена, значення, вказівники, змінні, константи, операції, вирази	
3,4	<u>Лекції 3-4. Базові поняття мови програмування С</u> Типи даних мови С. Алфавіт мови С. Лексичні одиниці мови С. Змінні, ініціалізація змінних, іменовані константи. Вирази, перетворення типів операндів у виразах, операції. Складність виразів (стор. 113-135 [1], стор. 18-20, 27-31, 181-191 [2], стор. 37-60 [3-укр], стор. 11-64 [4], стор. 6-14, 19-26 [5]).
3	<u>Лабораторна робота 2.</u> Програмування виразів. Введення-виведення даних
Тема 1.4. Базові структури програмування: послідовність, розгалуження, цикли	
5	<u>Лекція 5. Оператори мови програмування С. Програми лінійної структури. Програми з розгалуженнями</u> Оператор присвоювання. Оператор виклику функції; форматове введення й виведення даних. Блок операторів. Порожній оператор. Програми лінійної структури. Умовний оператор. Тернарна операція. Оператор вибору. Оператор безумовного переходу goto. Програми з розгалуженнями. (стор. 113-139, 143-163 [1], стор. 32-50, 192-198 [2], стор. 50-55,

	61-67, 173-190 [3], стор. 66-84, 86-94 [4], тор. 14-27, 42-43, 32-36 [5]; [8]).
5	<u>Лабораторна робота 3.</u> Розробка і реалізація алгоритмів розгалужених процесів з послідовною і вкладеною перевіркою умов (умовний оператор і оператор вибору).
6	<u>Лекція 6. Програмування циклічних процесів. Оператори циклів з передумовою і післяумовою. Цикли з параметрами</u> Поняття циклу. Циклічні конструкції. Оператор циклу з передумовою. Оператор циклу з післяумовою. Циклічні алгоритми введення інформації. Оператор циклу з параметром. Часткові нестандартні випадки використання циклу for. Оператори break і continue. Оптимізація циклічних алгоритмів. (стор. 167-222 [1], стор. 50-57, 64-76 [2], 67-73 [3], стор. 94-105 [4], стор. 36-42 [5]).
7	<u>Лекція 7. Розширені можливості роботи з цілими числами. Вказівники</u> Перераховувані дані. Побітові операції. Вказівники. (стор. 123-125, 247, 274-278, 355 [1], стор. 147-149, 168-172 [2], стор. 39 (41-42), 48-49 (52-54), 93-126 (105-144) [3], стор. 39-42, 48-51, 112-124 [4], стор. 24-25, 43-61 [5]).
7	<u>Лабораторна робота 4.</u> Проектування алгоритмів і програм циклічної структури. Цикли різних типів, вкладені цикли, робота з дійсними числами.
Тема 1.5. Процедурно-орієнтоване програмування. Рекурсія	
8	<u>Лекція 8. Модульне програмування. Функції в мові C</u> Модульне програмування. Оголошення і визначення функції. Порожній тип void. Виклик функції; організація міжмодульних зв'язків. Оператор return. Локальні, глобальні, статичні змінні. (стор. 229-270 [1], стор. 86-98 [2], стор. 75-96, 107-110 [3], стор. 183-194 [4], стор. 99-119 [5]).
9	<u>Лекція 9. Рекурсивні функції</u> Поняття рекурсії. Приклади рекурсивних алгоритмів. Оптимізація рекурсивних алгоритмів. (стор. 247-266 [1], стор. 98 [2], стор. 96-98 [3], стор. 222-227 [4], стор. 110-112 [5]).
9	<u>Лабораторна робота 5.</u> Наближене обчислення сум рядів. Розробка програм модульної структури.
Розділ 2. Структури даних і алгоритми	
Тема 2.1. Організація даних (масиви, рядки, структури) та алгоритми їхньої обробки	
10,11	<u>Лекції 10-11. Одновимірні масиви. Структури</u> Поняття структури даних. масиви. Основні можливості роботи з масивами. Робота з масивами у функціях. Типові алгоритми роботи з одновимірними масивами. Пошук елемента в масиві. Сортування масивів. Приклади алгоритмів з використанням одновимірних масивів. Оптимізація алгоритмів обробки одновимірних масивів. Структури; сортування масивів структур. (стор. 273-341, 321-335 [1], 102-113, 149-152, 199-202 [2], стор. 110-114, 145-164 [3], стор. 125-131, 164-175, 195-197 [4], стор. 61-66, 76-87 [5]).
11	<u>Лабораторна робота 6.</u> Обробка одновимірних масивів. Сортування в одновимірних масивах. Сортування масивів структур.
12	<u>Лекція 12. Двовимірні масиви</u> Багатовимірні масиви. Можливості роботи з двовимірними масивами. Багатовимірні масиви і функції. Типові алгоритми роботи з двовимірними масивами. Приклади алгоритмів з використанням двовимірних масивів. Розріджені матриці. Оптимізація алгоритмів обробки двовимірних масивів. (стор. 352-387 [1], стор. 123-134, 203-208 [2], 126-128 [3], стор. 131-140 [4], стор. 66-69 [5]).
13	<u>Лекція 13. Рядки символів.</u> Символьні дані. Рядки символів. Можливості роботи з рядками символів. Приклади обробки текстової інформації. (стор. 369-373 [1], стор. 138-149 [2], стор. 52-54 [3], стор. 48-51, 141-162 [4], стор. 71-76 [5]).

13	Лабораторна робота 7. Обробка двовимірних масивів. Робота з секторами матриці.
Тема 2.2. Алгоритмізація типових обчислювальних задач	
14	Лекція 14. Алгоритмізація розв'язування задач Вибір алгоритму розв'язування задачі. Генерація псевдовипадкових чисел. Генерація деяких комбінаторних множин (стор. 392-501 [1]; [6]).
15	Лекція 15. Модульна контрольна робота
16	Залік

6. Самостійна робота студента

Самостійна робота студента охоплює такі складники як підготовка до поточних опитувань, підготовка до лабораторних занять, модульної контрольної роботи й заліку. Протягом семестру після кожної лекції студентам для глибшого ознайомлення з матеріалом дисципліни надаються питання для виконання СРС. Лекційний матеріал і самостійно поглиблено опрацьовані студентом питання використовуються при виконанні лабораторних робіт.

Теми, які виносяться на самостійну роботу студентів:

Розділ 1. Організація програм

Тема 1.1. Архітектура комп'ютерів. Системи числення

Архітектура сучасних ПК. Подання цілих і дробових чисел в двійковій системі числення (стор. 78-94 [1], стор. 9-15 [2], стор. 1-52 [6], [9]).

Тема 1.2. Поняття алгоритму. Типові алгоритмічні структури

Типи обчислювальних процесів. Розвиток мов програмування (стор. 39-43 [1], стор. 73-89 [7]).

Тема 1.3. Елементи алгоритмічних мов: концепція типів даних, імена, значення, вказівники, змінні, константи, операції, вирази

Структура програми мовою C. Типи даних. Операції (стор. 113-135 [1], стор. 37-60 [3], стор. 32-38 [4]).

Тема 1.4. Базові структури програмування: послідовність, розгалуження, цикли

Оптимізація лінійних обчислювальних процесів (стор. 133-139 [1], стор. 7-13 [5]).

Оптимізація розгалужених алгоритмів (стор. 157-163 [1]).

Складність та оптимізація циклічних алгоритмів (стор. 218-222 [1], стор. 19 [5]).

Часткові нестандартні випадки використання циклу for (стор. 168-218 [1], 94-100 [4]).

Програмування ітераційних і вкладених циклів (стор. 168-218 [1], стор. 64-76 [2]).

Побітові (порозрядні) операції (стор. 123-125 [1], стор. 147-149 [2], 48-51 [4]).

Тема 1.5. Процедурно-орієнтоване програмування. Рекурсія

Складність та оптимізація алгоритмів модульної структури (стор. 266-270 [1]).

Розробка рекурсивних алгоритмів (стор. 98 [2], стор. 222-227 [4]).

Розділ 2. Структури даних і алгоритми

Тема 2.1. Організація даних (масиви, рядки, структури) та алгоритми їхньої обробки

Алгоритми сортування. Складність і оптимізація алгоритмів обробки одновимірних масивів (стор. 297-320, 335-341 [1])

Складність і оптимізація алгоритмів обробки двовимірних масивів (стор. 385-387 [1]).

Робота зі структурами (стор. 149-152 [2], стор. 145-164 [3], стор. 164-175 [4]).

Тема 2.2. Алгоритмізація типових обчислювальних задач

Генерація комбінаторних множин (стор. 453-489 [1]).

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування занять. Відвідування лекційних і лабораторних занять є обов'язковим за винятком поважних причин (хвороба, непередбачувані обставини).

У разі пропуску занять з поважних причин викладач надає можливість студенту виконати всі лабораторні завдання.

У разі пропуску занять без поважних причин, а також через порушення граничного терміну виконання завдання студент може отримати 80% від максимальної оцінки за відповідне завдання.

Протягом семестру студенти:

- виконують і захищають лабораторні роботи;
- пишуть модульну контрольну роботу;
- складають залік.

Пропущені контрольні заходи оцінювання. Кожен студент має право відпрацювати пропущені з поважної причини заняття за рахунок самостійної роботи.

Процедура оскарження результатів контрольних заходів оцінювання. Студенти мають право аргументовано оскаржити результати контрольних заходів, пояснивши, з яким критерієм не погоджуються.

Календарний контроль проводиться з метою підвищення якості навчання студентів та моніторингу виконання студентом вимог силабусу.

Академічна доброчесність. Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

Норми етичної поведінки. Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

Інклюзивне навчання. Засвоєння знань та умінь в ході вивчення дисципліни «Алгоритмізація та програмування. Частина 1. Базові концепції програмування» може бути доступним для більшості осіб з особливими освітніми потребами, крім здобувачів з серйозними вадами зору, які не дають можливості виконувати завдання з використанням персональних комп'ютерів, ноутбуків та/або інших технічних засобів.

Навчання іноземною мовою. У ході виконання завдань студентам може бути рекомендовано звернутися до англомовних джерел чи джерел іншими мовами Європейського Союзу.

Призначення заохочувальних та штрафних балів Відповідно до Положення про систему оцінювання результатів навчання сума всіх заохочувальних балів не може перевищувати 10% рейтингової шкали оцінювання.

Заохочувальні бали		Штрафні бали	
Критерій	Ваговий бал	Критерій	Ваговий бал
Написання тез, статті (за тематикою навчальної дисципліни)	2-4 бали	-	-
Участь у міжнародних, всеукраїнських та/або інших заходах та/або конкурсах (за тематикою навчальної дисципліни)	3-5 балів	-	-
Реалізація методу, який не вивчається в курсі	1-2 бали	-	-

Підготовка до лабораторних занять і контрольних заходів здійснюється під час самостійної роботи студентів з можливістю консультування з викладачем у визначений час консультацій або за допомогою електронних засобів (електронна пошта, телеграм, зум).

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

1. Поточний контроль (виконання робіт у семестрі) оцінюється в 100 балів. Розподіл балів подано в таблиці:

№ з/п	Контрольний захід оцінювання	Максимальна кількість балів за виконання однієї роботи	Разом
1	Виконання і захист лабораторної роботи 1-7	5	35
2	Звіт з лабораторної роботи 1-7	2	14
3	Поточна контрольна робота 1, 2	10	20
4	Наявність повного конспекту лекцій	6	6
5	Модульна контрольна робота	25	25
Разом			100

Поточні контрольні роботи містять по 5 завдань; максимальний бал за кожне завдання — 2. Максимальний ваговий бал — 2 бали × 5 = 10 балів.

Модульна контрольна робота складається з 5 завдань; максимальний бал за кожне завдання — 5. Максимальний ваговий бал — $5 \text{ балів} \times 5 = 25 \text{ балів}$.

Штрафні бали віднімаються за:

- 1) неоптимальний алгоритм — 10% від максимальної кількості балів;
- 2) неоптимальні структури подання інформації — 10% від максимальної кількості балів;
- 3) ненадану або невірну відповідь на запитання — 20% від максимальної кількості балів при захисті лабораторної роботи або 100% — на контрольній роботі.

2. Календарний контроль проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу:

<i>Критерій</i>	<i>Перший календарний контроль</i>	<i>Другий календарний контроль</i>
Термін календарного контролю	Тиждень 7-8	Тиждень 14-15
Умови отримання позитивної оцінки — поточний рейтинг	≥ 23 балів	≥ 40 балів

3. Умови допуску до заліку: відсутність заборгованостей з лабораторних робіт 1-7.

Студенти, які виконали умови допуску до заліку і мають рейтингову оцінку 60 і більше балів, отримують відповідну до набраного рейтингу оцінку без додаткових випробувань.

Зі студентами, які виконали умови допуску до заліку і мають рейтингову оцінку менше 60 балів, а також з тими студентами, хто бажає підвищити свою рейтингову оцінку, на останньому за розкладом занятті з дисципліни в семестрі проводиться семестровий контроль у вигляді заліку.

4. Відповідь на заліку оцінюється у 100 балів. Кожне завдання залікової роботи містить по 4 питання — 2 теоретичних і 2 практичних, кожне з яких має максимальний бал — 25. Максимальний ваговий бал — $25 \text{ балів} \times 4 = 100 \text{ балів}$.

Теоретична частина оцінюється так:

- правильна чітко викладена, повна відповідь (не менше 90% потрібної інформації) — 23-25 балів;
- достатньо повна відповідь (не менше 75% потрібної інформації) — 19-22 бали;
- неповна відповідь (не менше 60% потрібної інформації) — 15-18 балів;
- незадовільна відповідь — 0 балів.

Практичне завдання оцінюється так:

- повне безпомилкове розв'язування завдання — 23-25 балів;
- повне розв'язування завдання із несуттєвими невідповідностями — 19-22 бали;
- завдання виконано з певними недоліками — 15-18 балів;
- завдання не виконано — 0 балів.

5. Рейтингова оцінка за освітній компонент за бажанням студента визначається одним з таких способів:

- 1) кількість балів, отриманих за поточний контроль;
- 2) результат виконання залікової контрольної роботи (тоді не враховуються бали, отримані в семестрі).

Таблиця переведення рейтингових балів в оцінку за університетською шкалою:

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

Виставлення оцінки за контрольні заходи шляхом перенесення результатів проходження он-лайн-курсів чи інших курсів не передбачено.

9. Додаткова інформація з дисципліни

Перелік теоретичних питань, які виносяться на семестровий контроль (залік):

1. Структура й принципи роботи ПЕОМ.
2. Поняття алгоритму. Властивості алгоритму.
3. Подання алгоритмів. Словесне подання алгоритму (метасмова). Подання алгоритму за допомогою блок-схеми.
4. Типи обчислювальних структур: лінійні, розгалужені, циклічні.
5. Основні поняття програмування. Синтаксис, синтаксичні помилки. Семантика, семантичні помилки. Налагодження й тестування програм.
6. Методи трансляції програм: інтерпретація, компіляція.
7. Основні положення структурного програмування.
8. Модульне програмування. Переваги й недоліки.
9. Проектування й програмування згори вниз і знизу вгору.
10. Структурне програмування і коректність програм.
11. Системи числення. Двійкове, вісімкове, шістнадцяткове подання цілих і дробових чисел.
12. Особливості роботи з дійсними числами в комп'ютері.
13. Структура програми мовою C: декларативна й виконувана частини.
14. Лексичні одиниці мови C: ключові слова, ідентифікатори, константи.
15. Типи даних в мові C. Перетворення типів у виразах і при присвоюванні.
16. Введення й виведення даних (функції printf, scanf). Формати.
17. Запис цілих, дійсних, логічних, символьних констант мови C.
18. Оголошення й використання констант у мові C.
19. Змінні стандартних типів мови C: цілі, дійсні, символьні.
20. Операції мови C. Операція присвоювання.
21. Операції мови C. Операція кома.
22. Арифметичні вирази.
23. Оцінка складності виразів; оптимізація обчислень.
24. Операції порівняння. Логічні вирази.
25. Умовний оператор мови C. Вкладені умовні оператори.
26. Тернарна операція ?:
27. Оператор вибору мови C.
28. Оцінка обчислювальної складності алгоритмів з розгалуженнями.
29. Поняття циклу. Оператори циклу мови C. Вкладені цикли.
30. Оператор циклу з параметром. Схема виконання.
31. Нестандартні можливості використання оператора циклу з параметром.
32. Оператор циклу з передумовою. Схема виконання.
33. Оператор циклу з післяумовою. Схема виконання.
34. Складність алгоритмів з циклами; оптимізація виконання, застосування мемоізації.
35. Функції мови C: стандартні й описувані.
36. Функції мови C: з параметрами і без параметрів.
37. Функції, які повертають значення. Оператор return.
38. Локальні, глобальні й статичні змінні.
39. Фактичні й формальні параметри.
40. Механізми передачі параметрів за адресою і за значенням.
41. Оцінка складності модульного алгоритму.
42. Рекурсивні функції. Оцінка складності рекурсивного алгоритму.
43. Одновимірні масиви мови C: опис, ініціалізація, введення-виведення.
44. Методи сортування елементів масиву: обмінне сортування простою вибіркою з пошуком мінімуму чи максимуму.
45. Методи сортування елементів масиву: обмінне сортування (метод бульбашки) з індикатором перестановки.
46. Методи сортування елементів масиву: шейкер-сортування.
47. Методи сортування елементів масиву: сортування простими вставками.
48. Двовимірні масиви мови C: опис, ініціалізація, введення-виведення.
49. Передача параметрів-масивів у функції.
50. Типові алгоритми роботи з двовимірними масивами.
51. Сортування елементів матриці.
52. Робота з секторами матриці, обхід елементів матриці.
53. Розріджені матриці.

- 54. Змінні символного типу і рядки символів.
- 55. Основні функції роботи з рядками символів.
- 56. Перераховувані дані: опис, ініціалізація, використання.
- 57. Побітові операції.
- 58. Структури: опис, робота, передача в функції.
- 59. Масиви структур: опис, робота, передача в функції.
- 60. Використання випадкових чисел.

Робочу програму навчальної дисципліни (силабус)

склала: доцент кафедри цифрових технологій в енергетиці,
канд. техн .наук, доцент, Кублій Лариса Іванівна

Ухвалено кафедрою ЦТЕ (протокол № 22 від 25.06.2025)

Погоджено Методичною радою НН ІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 27.06.2025)