

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ

Декан факультету комп'ютерних
інформаційних технологій


Ігор ЯКИМЕНКО
" 29 " *серпень* 2025 р.

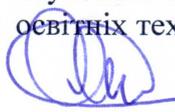
ЗАТВЕРДЖУЮ

Проректор з
науково-педагогічної роботи


Віктор ОСТРОВЕРХОВ
" 29 " *серпень* 2025 р.

ЗАТВЕРДЖУЮ

Директор навчально-
наукового інституту новітніх
освітніх технологій


Святослав ГРИТЕЛЬ
" 29 " *серпень* 2025 р.

РОБОЧА ПРОГРАМА

з дисципліни «Основи програмування»

Ступінь вищої освіти: бакалавр

Галузь знань – F «Інформаційні технології»

Спеціальність – F2 «Інженерія програмного забезпечення»

Освітньо-професійна програма – «Інженерія програмного забезпечення»

Кафедра комп'ютерних наук

| Форма навчання | Курс | Семестр | Лекції (год.) | ЛР (год.) | ІРС (год.) | Тренінг (год.) | СРС (год.) | Разом (год.) | Іспит (сем.) |
|----------------|------|---------|---------------|-----------|------------|----------------|------------|--------------|--------------|
| Денна | 1 | 1 | 30 | 30 | 4 | 8 | 78 | 150 | 1 |
| Заочна | 1 | 1, 2 | 8 | 4 | - | - | 138 | 150 | 2 |

Тернопіль – ЗУНУ
2025

29.08.2025


Робоча програма складена на основі освітньо-професійної програми підготовки бакалавра галузі знань F «Інформаційні технології» спеціальності F2 «Інженерія програмного забезпечення», затвердженої Вченою Радою ЗУНУ (протокол № 8 від 26 червня 2025 р.).

Робочу програму склав к.т.н., доцент

Олег ПІЦУН

Робоча програма затверджена на засіданні кафедри комп'ютерних наук, протокол №1 від 26 серпня 2025 р.

Завідувач кафедри, д.т.н., професор



Андрій ПУКАС

Гарант ОП,
к.т.н., доцент



Світлана КРЕПИЧ

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

"Основи програмування"

1. Опис дисципліни "Основи програмування"

| Дисципліна - Основи програмування | Галузь знань, спеціальність, ступінь вищої освіти | Характеристика навчальної дисципліни |
|--|--|--|
| Кількість кредитів – <i>Денна – 5,</i> <i>Заочна – 5</i> | Галузь знань: F «Інформаційні технології» | Статус дисципліни: Обов'язкова Мова навчання: українська |
| Кількість залікових модулів - 5 | Спеціальність: F2 «Інженерія програмного забезпечення»; | Рік підготовки: денна – 1; заочна – 1. Семестр: денна – 1; заочна – 1,2. |
| Кількість змістовних модулів – <i>Денна – 4</i> <i>Заочна - 4.</i> | Ступінь вищої освіти: бакалавр | Лекції: денна – 30 год.; заочна – 8 год. Лабораторні роботи: денна –30 год.; заочна – 4 год. |
| Загальна кількість годин - 150 | | Самостійна робота: денна – 78 год.; тренінг - 8. заочна – 138 год. Індивідуальна робота: денна – 4 год. |
| Тижневих годин <i>Денна: 1 семестр - 10 год., з</i> <i>них аудиторних – 4 год.</i> | | Вид підсумкового контролю <i>Денна: 1 семестр – екзамен</i> <i>Заочна: 2 семестр – екзамен</i> |

2. Мета й завдання вивчення дисципліни

"Основи програмування"

2.1. Мета вивчення дисципліни

Програма та тематичний план дисципліни орієнтовані на глибоке та ґрунтовне засвоєння студентами основних алгоритмів розв'язку типових задач та їх реалізація за допомогою конкретних алгоритмічних мов програмування. Ця дисципліна відноситься до фундаментальних наук, які формують фаховий світогляд майбутніх програмістів.

Студенти вивчають теоретичні та практичні аспекти розробки програмного забезпечення з використанням сучасних підходів.

Курс "Основи програмування" охоплює методологічні основи алгоритмізації, методи та прийоми вивчення інформаційних систем, їх моделювання та реалізацію тощо. Курс повинен сприяти формуванню висококваліфікованих фахівців у галузі інформаційних технологій.

2.2. Завдання вивчення дисципліни

Головним завданням дисципліни є набуття студентами теоретичних знань про принципи побудови сучасних мов програмування, вивчення основних конструкцій мов, типових задач алгоритмічного програмування, набуття досвіду роботи в інтегрованому середовищі розробки

програм (на прикладі компілятора Visual Studio C++) та проходженні повного циклу розробки (розробка алгоритму - кодування - компіляція - відлагодження - документування). Таке поєднання теоретичних та практичних знань та вмінь сприяє як достатньому оволодінню роботою з конкретними існуючими середовищами розробки, так і полегшенню освоєння нових засобів програмування та переходу на нові середовища та операційні системи в майбутньому.

Вивчення курсу "Основи програмування" передбачає наявність систематичних та ґрунтовних знань отриманих при вивченні шкільного курсу "Основи інформатики", цілеспрямованої роботи над вивченням спеціальної літератури, активної роботи на лекціях, практичних та лабораторних заняттях, самостійної роботи та виконання індивідуальних завдань.

2.3 Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни

K01. Здатність до абстрактного мислення, аналізу та синтезу;

K02. Здатність застосовувати знання у практичних ситуаціях;

K15. Здатність розробляти архітектури, модулі та компоненти програмних систем;

K22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя;

K23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення;

K24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення;

K25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення;

K26. Здатність до алгоритмічного та логічного мислення.

2.4. Передумови для вивчення дисципліни.

Зазначена дисципліна включена до циклу дисциплін професійної підготовки за переліком програми. У структурно-логічній схемі навчання зазначена дисципліна розміщена на I-му курсі. Вивчення курсу "Основи програмування" передбачає наявність систематичних та ґрунтовних знань із курсу «Інформатика» середньої школи, цілеспрямованої роботи над вивченням спеціальної літератури, активної роботи на лекціях та практичних заняттях, самостійної роботи.

2.5. Результати навчання

В результаті вивчення дисципліни студенти повинні:

ПР03 Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПР13 Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань

ПР15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

3. Програма навчальної дисципліни "Основи програмування"

Змістовний модуль 1. Основні поняття програмування.

Тема 1. Основні поняття та означення

Поняття архітектури комп'ютера. Архітектури комп'ютера фон Неймана. Архітектура системи команд. Інформація в пам'яті комп'ютера. Типи комп'ютерів. Програмне забезпечення. Засоби створення програм. Поняття алгоритму. Властивості алгоритмів, форми їх представлення. Основні структури алгоритмів.

Тема 2. Синтаксис мови C++

Множина символів C++ та множина представимих символів. Правила формування констант, ідентифікаторів. Ключові слова. Використання коментарів. Поняття лексеми.

Змістовний модуль 2. Основні структури в програмуванні.

Тема 3. Структура програми на мові C++

Склад програми на мові C++. Вихідні файли програми. Виконання програм. Особливості функції `main()`. Поняття "час життя" і "область видимості". Приклади найпростіших програм.

Тема 4. Оголошення змінних

Базові типи даних. Правила приведення базових типів даних. Модифікатори змінних. Автоматичні змінні. Реєстрові змінні. Зовнішні змінні та функції статичні змінні. Змінні класу `volatile`. Новий стиль заголовків.

Тема 5. Керування вводом-виводом на екран

Основні оператори вводу та виводу на мові C++, зокрема на прикладі `printf()` та `scan()`. Формат вводу-виводу різних типів даних (чисел, рядків, вказівників тощо).

Тема 6. Вирази

Арифметичні операції. Оператор присвоювання. Поняття виразу. Оператори інкремента і декремента. Оператор `sizeof`. Порозрядні логічні операції. Операції зсуву вліво і вправо. Оператори порівняння. Операція "кома". Пріоритет і порядок виконання операцій.

Змістовний модуль 3. Засоби реалізації основних структур в програмуванні .

Тема 7. Засоби мови C++ для реалізації основних структур алгоритмів

Умовні оператори. Оператори `if`. Оператори `if-else`. Умовний оператор `?:`. Оператор `switch`. Оператори циклу. Цикли `for`. Цикли `while`. Цикли `do-while`. Оператор `break`. Оператор `continue`. Оператор `goto` і метки.

Тема 8. Складні типи даних в C++

Масиви. Ініціалізація масивів. Багатомірні масиви. Масиви як параметри функцій. Структури й операції з ними. Структури як аргументи функцій. Масиви структур. Показники на структури. Передача по посиланню членів масивів структур. Об'єднання й операції з ними.

Тема 9. Вказівники та посилання

Загальний огляд. Арифметика вказівників. Вказівники. на вказівники. Вказівники. на функції. Посилання. Передача параметрів за посиланням і за значенням. Використання вказівників і посилань із ключовим словом `const`.

Змістовний модуль 4. Функції роботи зі стрічками та ООП .

Тема 10. функції

Параметри й аргументи функцій. Аргументи за замовчуванням. Простір імен. Функції, що вбудовуються (`inline`-). Рекурсивні функції. Математичні функції. Функції округлення.

Тема 11. Об'єктно-орієнтоване програмування

Класи. Методи. Наслідування. Інкапсуляція. Поліморфізм. Властивості та поведінка класів.

Тема 12. Бібліотека роботи з графічними елементами SFML.

Поняття технології SFML. Принципи підключення функцій бібліотеки SFML в проект. Створення візуального вікна для відображення графічних елементів. Відображення графічних примітивів за допомогою функцій бібліотеки SFML.

Тема 13. Файловий ввід / вивід

Взаємодія з файлами. Типи файлів. Функції стандартного вводу / виводу. Функції довільного доступу до файлу.

4. Структура залікового кредиту дисципліни "основи програмування"

| денна форма навчання | Кількість годин | | | | | | Контрольні заходи |
|---|-----------------|-----|-------|----|-----|---------|--------------------------------|
| | Лекції | Лаб | Практ | СР | ІРС | Тренінг | |
| Змістовний модуль 1. Основні поняття програмування | | | | | | | |
| Тема 1. Основні поняття та означення | 2 | 2 | | 6 | 1 | 2 | Усне опитування/ тестування |
| Тема 2. Синтаксис мови C++ | 2 | 2 | | 6 | | | Усне опитування/ тестування |
| Змістовний модуль 2. Основні структури в програмуванні | | | | | | | |
| Тема 3. Структура програми на мові C++ | 2 | 2 | | 6 | 1 | 2 | Усне опитування/ тестування |
| Тема 4. Оголошення змінних | 2 | 2 | | 6 | | | Усне опитування/ тестування |
| Тема 5. Керування вводом-виводом на екран | 2 | 2 | | 6 | | | Усне опитування/ тестування |
| Тема 6. Вирази | 2 | 2 | | 6 | | | Усне опитування/ тестування |
| Змістовний модуль 3. Засоби реалізації основних структур в програмуванні | | | | | | | |
| Тема 7. Засоби мови C++ для реалізації основних структур алгоритмів | 2 | 2 | | 6 | 1 | 2 | Усне опитування/ тестування |
| Тема 8. Складні типи даних в C++ | 2 | 2 | | 6 | | | Усне опитування/ тестування |
| Тема 9. Вказівники та посилання | 2 | 2 | | 6 | | | Усне опитування/ тестування |
| Змістовний модуль 4. Функції роботи зі стрічками та ООП | | | | | | | |
| Тема 10. функції | 4 | 4 | | 6 | 1 | 2 | Усне опитування/ тестування |
| Тема 11. Об'єктно-орієнтоване програмування | 4 | 4 | | 6 | | | Усне опитування/ тестування |
| Тема 12. Бібліотека роботи з графічними елементами SFML. | 2 | 2 | | 6 | | | Усне опитування/ тестування |
| Тема 13. Файловий ввід / вивід | 2 | 2 | | 6 | | | Усне опитування/ тестування |
| Разом | 30 | 30 | - | 78 | 4 | 8 | |

| заочна форма навчання | Кількість годин | | |
|---|-----------------|---------------------|-------------------|
| | Лекції | Лабораторні заняття | Самостійна робота |
| Змістовний модуль 1. Основні поняття програмування | | | |
| Тема 1. Основні поняття та означення | 2 | 1 | 7 |
| Технологічний цикл розробки програм | | | 7 |
| Тема 2. Синтаксис мови C++ | | | 7 |
| Написання програм в середовищі MS Visual Studio. | | | 7 |
| Змістовний модуль 2. Основні структури в програмуванні | | | |
| Тема 3. Структура програми на мові C++ | 2 | 1 | 7 |
| Тема 4. Оголошення змінних | | | 7 |

| | | | |
|---|---|---|-----|
| Програмування простих задач | | | 8 |
| Тема 5. Керування вводом-виводом на екран | | | 8 |
| Адресна арифметика | | | 8 |
| Тема 6. Вирази | | | 8 |
| Арифметичні вирази у мові C/C++ | | | 8 |
| Програмування циклічних процесів | | | 8 |
| Змістовний модуль 3. Засоби реалізації основних структур в програмуванні | | | |
| Тема 7. Засоби мови C++ для реалізації основних структур алгоритмів | | | 8 |
| Логічні вирази у мові C/C++ | | | 8 |
| Побітові вирази у мові C/C++ | | | 8 |
| Тема 8. Складні типи даних в C++ | 2 | 1 | 8 |
| Алгоритми з циклічною структурою | | | 8 |
| Робота з масивами | | | 8 |
| Тема 9. Вказівники та посилання | | | 8 |
| Робота з даними типу "структура" | | | 8 |
| Змістовний модуль 4. Функції та робота з строковими змінними | | | |
| Тема 10. функції | | | 8 |
| Використання функцій. | | | 8 |
| Тема 11. Об'єктно-орієнтоване програмування | | | 8 |
| Тема 12. Бібліотека роботи з графічними елементами SFML. | 2 | 1 | 8 |
| Обробка стрічкових даних | | | 8 |
| Тема 13. Файловий ввід / вивід | | | 8 |
| Ввід-вивід у файл. | | | 8 |
| Разом | 8 | 4 | 138 |

5. Тематика лабораторних робіт

Лабораторна робота №1

Тема: Написання програм в середовищі VS C++

Мета: Навчитись працювати в середовищі програмування VS C++.

Лабораторне заняття №2

Програмування простих задач

Мета: Навчитися складати алгоритми та програмувати задачі невисокої складності

Лабораторне робота №3

Програмування циклічних процесів

Мета: Навчитися складати алгоритми та програмувати задачі з циклічними процесами

Лабораторне робота №4

Робота з масивами

Мета: Навчитися складати алгоритми та програмувати задачі, що використовують масиви.

Лабораторне робота №5

Робота з даними типу "структура"

Мета: Навчитися складати алгоритми та програмувати задачі елементами даних «структура».

Лабораторне робота №6

Використання функцій

Мета: Навчитися складати алгоритми та програмувати задачі з використанням функцій

користувача.

Лабораторне робота №7

Об'єктно – орієнтоване програмування

Мета: Навчитися розробляти програми з використанням класів та базових парадигм об'єктно – орієнтованого програмування.

Лабораторне робота №8

Ввід-вивід у файл

Мета: Навчитися складати алгоритми та програмувати задачі з файловими потоками

6. Самостійна робота

Для успішного вивчення і засвоєння дисципліни «Основи програмування» студенти повинні володіти значним обсягом інформації, частину якої вони отримують і опрацьовують шляхом самостійної роботи. Самостійна робота полягає в опрацюванні сучасної навчальної і наукової фахової літератури.

Результати самостійної роботи студента з дисципліни «Основи програмування» оцінюються за 100-бальною шкалою у вигляді аналітичного огляду наукових джерел або літератури та програмної реалізації відповідно до обраної з переліку теми або запропонованої власної теми.

1. Алгоритм Евкліда . Реалізувати програму для знаходження найбільшого спільного дільника двох чисел за допомогою алгоритму Евкліда.
2. Сортування вставками . Написати програму для сортування масиву методом вставок.
3. Метод підрахунку . Реалізувати програму для сортування масиву методом підрахунку.
4. Нерекурсивне обчислення числа Фібоначчі . Написати програму, що обчислює числа Фібоначчі без рекурсії.
5. Пошук найбільшого спільного кратного (НСД) . Реалізувати програму для знаходження НСК двох чисел.
6. Числа Армстронга . Написати програму для перевірки, чи є задане число числом Армстронга.
7. Числа щасливі та нещасливі . Реалізувати програму для перевірки, чи є число "щасливим" (наприклад, 7).
8. Сортування Шелла . Написати програму для сортування масиву за допомогою алгоритму сортування Шелла.
9. Реалізація алгоритму Лемера . Створити програму для генерації псевдовипадкових чисел за допомогою алгоритму Лемера.
10. Алгоритм сортування методом злиття . Реалізувати програму для сортування масиву методом злиття.
11. Генерація простих чисел (решето Ератосфена) . Написати програму для знаходження всіх простих чисел до заданого числа.
12. Ітеративна та рекурсивна реалізація обчислення факторіалу . Написати дві версії програми для обчислення факторіалу.
13. Аналіз алгоритму пошуку мінімального елемента в масиві . Реалізувати та порівняти різні методи пошуку мінімуму.
14. Реалізація кільцевого буфера . Написати програму для реалізації кільцевого буфера в масиві.

Ці теми більше орієнтовані на алгоритми, структури даних та нестандартні задачі, що можуть бути корисними для поглиблення знань з C++ за допомогою вказівників.

15. Динамічна пам'ять . Створення програми для динамічного виділення пам'яті під масив.
16. Класи та об'єкти . Створення класу для опису студента (ім'я, прізвище, оцінки).
17. Конструктори та деструктори . Написати клас для управління масивом з конструктором і деструктором.
18. Наслідування . Розробити базовий клас "Транспортний засіб" та похідні класи "Автомобіль",

"Мотоцикл".

19. Поліморфізм . Реалізувати поліморфізм для обчислення площі різних фігур (коло, квадрат).
20. Віртуальні функції . Написати програму з використанням віртуальних функцій для наслідуваних класів.
21. Абстрактні класи . Створити абстрактний клас "Тварина" та реалізувати похідні класи "Кіт", "Собака".
22. Шаблони функцій . Написати шаблон функції для обчислення мінімуму та максимуму двох елементів.
23. Шаблони класів . Реалізувати шаблон класу для збереження різних типів даних.
24. Обробка файлів . Створити програму для читання даних з файлу та їх обробки.
25. Запис у файл . Написати програму для запису даних у файл.
26. Ведення журналу (логування) у файл . Реалізувати програму для ведення логів у текстовому файлі.
27. Структури . Створити структуру для опису автомобіля (марка, рік випуску, ціна) та реалізувати пошук автомобіля з мінімальною ціною.
28. Перерахування (enum) . Використання перерахувань для створення програми вибору типу меню.
29. Об'єктно-орієнтоване програмування (ООП): інкапсуляція . Створити клас "Банк", що інкапсулює інформацію про баланс рахунку.
30. ООП: композиція . Реалізувати клас "Комп'ютер", що складається з класів "Процесор", "Пам'ять", "Диск".
31. ООП: агрегація . Створити клас "Факультет", що містить студентів як об'єкти.
32. ООП: наслідування із захищеними членами . Реалізувати клас "Співробітник" і похідні "Менеджер", "Програміст".
33. Робота з бібліотеками STL: вектори . Написати програму для сортування елементів вектора.
34. STL: списки . Реалізувати програму для додавання та видалення елементів зі списку.
35. STL: черги . Створити програму для симуляції черги клієнтів у банку.
36. STL: стеки . Реалізувати стек для перевірки коректності вкладених дужок у математичному виразі.
37. STL: множини та асоціативні контейнери . Використати множини для видалення дубльованих елементів із масиву.
38. STL: мапи . Створити мапу для збереження даних про студентів (ID - ім'я) та пошуку за ID.
39. Алгоритми STL . Написати програму для пошуку мінімального елемента в контейнері з використанням STL.
40. Робота з потоками введення-виведення . Реалізувати програму для роботи з двома потоками вводу та виводу.
41. Прототипування та тестування коду . Створення програми для тестування простих математичних функцій.
42. Обробка винятків . Реалізувати програму, яка перевіряє ділення на нуль та обробляє помилку.
43. Множинне наслідування . Написати клас "Людина", який наслідується від класів "Студент" та "Співробітник".
44. Алгоритм сортування бульбашкою . Реалізувати алгоритм сортування масиву за зростанням.
45. Алгоритм сортування вибором . Написати програму для сортування масиву за допомогою методу вибору.
46. Алгоритм сортування швидким методом (Quicksort) . Реалізувати алгоритм швидкого сортування.
47. Бінарний пошук . Створити програму для пошуку елемента в відсортованому масиві за допомогою бінарного пошуку.
48. Робота з рекурсією: Ханойська вежа . Реалізувати алгоритм рішення Ханойської вежі.
49. Графи: пошук в глибину . Написати програму для реалізації алгоритму пошуку в глибину.
50. Графи: пошук в ширину . Реалізувати програму для обходу графа методом пошуку в ширину.

7. Тренінг з дисципліни

Тематика: Проектування та реалізація програмних продуктів.

Порядок проведення:

1. Здійснити аналіз отриманого завдання відповідно до варіанту.
2. На основі аналізу побудувати алгоритм рішення задачі.
3. За алгоритмом створити код програми та реалізувати його на певному засобі програмування.
4. Скласти тестові приклади та протестувати вирішену задачу відповідно до індивідуального завдання.

Індивідуальні варіанти:

В завданні 1 необхідно написати програму, яка виконує вказані операції (кожну операцію оформити окремою функцією) з інформацією, що знаходиться у текстовому файлі input. і записує всі результати роботи програми у файл output. Скласти блок-схему для алгоритму розв'язку задачі.

Вхідний файл : input.txt

Вихідний файл : output.txt

В завданні 2 необхідно організувати файл даних з вказаною нижче структурою та передбачити функції, які дозволяють:

- коригування обраного запису файлу;
- пошук інформації за різними полями;
- додавання записів у кінець бази даних;
- вилучення інформації з бази даних.

Варіант 1

1. Написати програму, яка копіює вміст вхідного файлу у вихідний файл; підраховує кількість чисел у тексті (не цифр, а саме чисел); виділяє всі слова, що складаються тільки з латинських літер; видаляє кожне друге слово.
2. Структура з ім'ям SKLAD, яка містить наступні поля:
NAME - назва товару.
TYPE - одиниця виміру товару:
QUANTITY - кількість одиниць товару:
COST - Ціна одиниці товару:

Варіант 2

1. Написати програму, яка копіює вміст вхідного файлу у вихідний файл: підраховує кількість слів у тексті, які починаються з голосної літери: знаходить всі слова, які містять непарну кількість приголосних літер: видаляє всі числа з тексту.
2. Структура з ім'ям ABITURIENT, яка містить наступні поля:
NAME - прізвище, ініціали:
GENDER - стать:
SPEC - назва спеціальності:
EXAM - результати вступних іспитів з трьох предметів (масив з трьох елементів).

Варіант 3

1. Написати програму, яка копіює вміст вхідного файлу у вихідний: інвертує рядок, подаючи його у зворотному вигляді: підраховує кількість чисел у тексті: видаляє всі слова, що починаються з голосних літер.
2. Структура з ім'ям TOVAR, яка містить наступні поля:
NAME - назва товару.
TYPE - одиниця виміру товару:
SORT - сорт товару.
QUANTITY - кількість одиниць товару:
COST - Ціна одиниці товару:

Варіант 4

1. Написати програму, яка копіює вміст вхідного файлу у вихідний: замінює всі великі літери, що входять до тексту на відповідні малі: визначає найдовше слово: видаляє всі слова, що містять непарну кількість приголосних літер.

2. Структура 3 ім'ям ITINERARY. яка містить наступні поля:
FIRST - назва початкового пункту маршруту.
FINAL - назва кінцевого пункту маршруту.
NUM - номер маршруту.
DISTANCE - відстань у кілометрах.

Варіант 5

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: підраховує кількість слів, які мають непарну довжину. виводить на екран частоту входження кожної Літери у тексті: перевіряє правильність розташування круглих дужок у тексті: видаляє всі парні числа з тексту.
2. Структура 3 ім'ям ABONENT. яка містить наступні поля:
NAME - прізвище абонента:
INIT - ініціали абонента:
NOMER - номер телефону:
ADRESS - домашня адреса.

Варіант 6

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: перевіряє. чи співпадає кількість відкритих і закритих дужок у введеному рядку (перевірити для круглих та квадратних дужок): знаходить найдовше слово: видаляє всі слова. що складаються тільки з латинських літер.
2. Структура з ім'ям AEROFLOT. яка містить наступні поля:
NAZV' - назва пункту призначення:
NUMR - номер літака:
TYPE - тип літака:
TIME - час відправлення.

Варіант 7

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: підраховує кількість слів непарної довжини: замінює всі слова. записані кирилицею на аналогічні. записані латиницею (обробка obrobka): видаляє всі слова. які починаються і закінчуються на голосну літеру.
2. Структура з ім'ям ABONENT. яка містить наступні поля:
NAME - прізвище та ініціали користувача мережі:
LOGIN - обліковий запис:
PASSWORD - пароль:
TYPE - тип облікового запису:
DATE - рік та місяць прийняття на роботу.

Варіант 8

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: підраховує кількість різних слів. що входять до заданого тексту: визначає кількість використаних символів: видаляє всі слова, що мають подвоєні літери.
2. Структура з ім'ям STUDENT. яка містить наступні поля:
NAME - прізвище та ініціали:
DATABIRTH - дата народження:
GROUP - номер групи:
SES - успішність з п'яти предметів (масив з п'яти елементів).

Варіант 9

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: підраховує кількість слів у тексті: виділяє слово. що містить найбільшу кількість голосних літер: видаляє з тексту всі непотрібні пробіли.
2. Структура з ім'ям NOTE. яка містить наступні поля:
NAME - прізвище. ім'я:
TEL - шестизначний номер телефону".
BDAY - день народження (масив із трьох чисел).

Варіант 10

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: підраховує кількість розділових знаків у тексті: виводить всі слова. що мають парну кількість літер: міняє місцями

першу і останню літери кожного слова.

2. Структура з ім'ям DETAL. яка містить наступні поля:

NAME - назва деталі:

SORT - сорт виробу.

DATE - дата виготовлення (масив із трьох чисел).

QUANT - кількість:

COST - Ціна одиниці.

Варіант 11

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: підраховує кількість цифр у тексті: визначає слова. що починаються з приголосних літер: знищує всі слова. які починаються і закінчуються за одну й ту ж літеру.

2. Структура з ім'ям TOVAR. яка містить наступні поля:

NAME - назва товару.

TYPE - одиниця виміру товару:

SORT - сорт товару.

QUANTITY - кількість одиниць товару:

COST - Ціна одиниці товару.

Варіант 12

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: підраховує кількість великих літер у тексті: визначає слова. що мають найменшу кількість літер: видаляє всі слова. що починаються з малої літери.

2. Структура з ім'ям TRAIN. яка містить наступні поля:

NAZV' - назва пункту призначення:

NUMR - номер поїзда:

DATE - дата відправлення:

TIME - час відправлення.

Варіант 13

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: підраховує кількість слів. які містять однакову кількість голосних і приголосних літер: визначає найдовше слово: видаляє з тексту всі слова-паліндроми.

2. Структура з ім'ям SKLAD. яка містить наступні поля:

NAME - назва товару.

TYPE - одиниця виміру товару:

QUANTITY - кількість одиниць товару:

COST - Ціна одиниці товару:

Варіант 14

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: виводить на екран всі символи. які розташовані після першого символу «:» підраховує кількість речень. що містять непарну кількість слів: видаляє з тексту всі слова. які розташовані після ком.

2. Структура з ім'ям ITINERARY. яка містить наступні поля:

FIRST - назва початкового пункту маршруту.

FINAL - назва кінцевого пункту маршруту.

NUM - номер маршруту.

DISTANCE - відстань у кілометрах.

Варіант 15

1. Написати програму. яка копіює вміст вхідного файлу у вихідний: підраховує кількість слів у тексті. які закінчуються на голосну літеру: знаходить всі слова. довжина яких менша п'яти символів: видаляє всі слова. які містять хоча б одну латинську літеру.

2. Структура з ім'ям TRAIN. яка містить наступні поля:

NAZV' - назва пункту призначення:

NUMR - номер поїзда:

DATE - дата відправлення:

TIME - час відправлення.

8. Засоби оцінювання та методи демонстрування результатів навчання.

У процесі вивчення дисципліни «Основи програмування» використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- стандартизовані тести;
- поточне опитування;
- залікове модульне тестування та опитування;
- презентації результатів виконаних завдань та досліджень;
- студентські презентації та виступи на наукових заходах;
- тренінги;
- екзамен.

9. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Основи програмування» визначається як середньозважена величина, залежно від питомої ваги кожної складової залікового кредиту:

| Модуль 1 | | Модуль 2 | | Модуль 3 | Модуль 4 | Модуль 5 |
|--|---|--|--|---|--|---|
| 10 % | 10 % | 10 % | 10 % | 5 % | 15% | 40 % |
| Поточне оцінювання | Модульний контроль 1 | Поточне оцінювання | Модульний контроль 2 | Тренінг | Самостійна робота | Екзамен |
| Середнє арифметичне із оцінок, отриманих за виконання лабораторних робіт 1-4 | Оцінка за результатами тестування на основі тем 1-2 залікового модулю (25 тестів по 4 бали) | Оцінка за поточне оцінювання визначається, як середнє арифметичне з оцінок, отриманих під час занять | Середнє арифметичне із оцінок, отриманих за виконання лабораторних робіт 5-8 | Визначається як єдина оцінка, отримана під час презентації проекту, розробленого під час тренінгу | Визначається як єдина оцінка по 100-бальній шкалі за результат самостійної роботи за обраною тематикою | Визначається як єдина оцінка по 100-бальній шкалі та складається з двох теоретичних питань по 25 балів і задачі – 50 балів. |

Оцінка за «Поточне оцінювання» визначається як середнє арифметичне оцінок, отриманих студентом при виконанні лабораторних завдань. Кожне завдання оцінюється за 100 бальною шкалою на основі якості виконання та відповідності поставленим завданням. Бали нараховуються відповідно до таких критеріїв:

- 15 балів – розуміння теоретичних відомостей (розкриття понятійного апарату відповідно до теми лабораторного заняття);
- 25 балів – правильність розв'язування (коректність реалізації, правильний вибір підходу);
- 10 балів – якість виконання (унікальність рішення, стиль програмування, якість програмного коду);
- 10 балів – результативність / тестування (програма видає коректний результат при тестуванні);
- 10 балів – аналіз і висновки (опис отриманих результатів, засоби їх досягнення);
- 30 балів – захист лабораторної роботи (відповідь на запитання викладача).

Модульний контроль відбувається шляхом проведення тестування на основі залікових модулів 1-2 (1 модульний контроль) та 3-4 (2 модульний контроль). Тестування складається з 25 тестів, кожне питання оцінюється у 4 бали.

Тренінг оцінюється з повнотою реалізації, якістю програмного коду, тестуванням та аналізом ефективності виконання завдань. Оцінювання рівня виконання завдань тренінгу проводиться викладачем за 100-бальною шкалою згідно таблиці 2.

Таблиця 2 – Критерії оцінювання тренінгу

| Завдання | Макс. | Високий рівень | Достатній | Середній | Низький |
|----------|-------|----------------|-----------|----------|---------|
|----------|-------|----------------|-----------|----------|---------|

| | | | | | |
|---|-----|--|--|---|-------------------------------|
| | бал | | рівень | рівень | рівень |
| 1. Здійснити аналіз отриманого завдання | 15 | 22-30 аргументоване, з порівняльним аналізом | 15-21 достатньо аргументоване, але без альтернатив | 10-14 поверхневий аналіз | 0-9 відсутнє або невідповідне |
| 2. На основі аналізу побудувати алгоритм рішення задачі. | 15 | 22-30 реалізація у повній мірі | 15-21 реалізація з незначними недоліками | 10-14 передбачено лише часткові випадки | 0-9 відсутнє або невідповідне |
| 3. За алгоритмом створити код програми та реалізувати його на певному засобі програмування. | 40 | 26-40 реалізація коду в повному виді | 17-25 неповна реалізація | 10-16 часткова відповідність, наявність помилок | 0-9 відсутнє або невідповідне |
| 4. Скласти тестові приклади та протестувати вирішену задачу 5. | 30 | 26-40 комплексне тестування | 17-25 тестування, часткове | 10-16 часткова відповідність, наявність помилок | 0-9 відсутнє або невідповідне |
| Разом | 100 | - | - | - | - |

Модуль «Самостійна робота» оцінюється викладачем за якістю виконаного індивідуального завдання та звіту відповідно до критеріїв, визначених у розділі 6 «Самостійна робота» цієї робочої програми. Підсумковий бал розраховується як сума оцінок за окремими критеріями у таблиці 3.

Таблиця 3 – Критерії оцінювання самостійної роботи

| Завдання | Макс. бал | Високий рівень | Достатній рівень | Середній рівень | Низький рівень |
|---|-----------|--|---|---|-------------------------------|
| Проектування алгоритму відповідно до завдання | 30 | 22-30 алгоритм реалізовано у повній мірі | 15-21 алгоритм реалізований в неповній мірі | 10-14 поверхневий аналіз | 0-9 відсутнє або невідповідне |
| Розробка модулів та зв'язків між ними для реалізації завдання | 30 | 22-30 універсальний підхід | 15-21 загальний, не універсальний підхід | 10-14 типові рішення | 0-9 відсутнє або невідповідне |
| Програмна реалізація завдання із тестування та документацією | 40 | 26-40 комплексне тестування | 17-25 тестування, часткове | 10-14 часткова відповідність, наявність помилок | 0-9 відсутнє або невідповідне |
| Разом | 100 | - | - | - | - |

Модуль екзамен оцінюється за 100-бальною шкалою та включає два теоретичних питання по 25 балів і задачу – 50 балів. Якщо студент розв'язує задачу, аналізує отриманий результат та обґрунтовує рішення, отримує – 50 балів. Розв'язана задача без аналізу та висновків – 40 балів. Отримано розв'язок без наведених схем та ґрунтового аналізу – 30 балів. Частковий розв'язок задачі – 20 балів. Частковий розв'язок задачі із неточностями – 10 балів.

Оцінювання теоретичної частини проводиться таким чином:

1-10 балів – у мінімальному обсязі наведено поверхневу відповідь на питання, без конкретизації;

11-18 балів – наведено деталізовану відповідь на питання, без прикладів реалізації;

19-25 балів – повна відповідь на питання з прикладом програмної реалізації.

Максимальна кількість балів за кожне з питань – 25 балів.

Отримана оцінка за національною шкалою ECTS виставляється відповідно до шкали оцінювання рівня знань.

Шкала оцінювання:

| За шкалою Університету | За національною шкалою | За шкалою ECTS |
|------------------------|------------------------|---|
| 90-100 | відмінно | A (відмінно) |
| 85-89 | добре | B (дуже добре) |
| 75-84 | | C (добре) |
| 65-74 | задовільно | D (задовільно) |
| 60-64 | | E (достатньо) |
| 35-59 | незадовільно | FX (незадовільно з можливістю повторного складання) |
| 1-34 | | F (незадовільно з обов'язковим повторним курсом) |

10. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

| № | Найменування | Номер теми |
|---|--|------------|
| 1 | Мультимедійний проектор | 1-13 |
| 2 | Проекційний екран | 1-13 |
| 3 | Операційна система Windows, наявність доступу до мережі Internet | 1-13 |
| 4 | Персональні комп'ютери | 1-13 |
| 5 | Комунікаційне програмне забезпечення (Zoom) для проведення занять у режимі онлайн (за необхідності) | 1-13 |
| 6 | Комунікаційна навчальна платформа (Moodle) для організації дистанційного навчання (за необхідності) | 1-13 |
| 7 | Базове програмне забезпечення Microsoft Office | 1-13 |
| 8 | Спеціалізоване програмне забезпечення для моделювання та програмування: Microsoft Visio, MS Visual Studio. https://cpp.sh/ - онлайн-додаток для компіляції програм на мові програмування C++. | 1-13 |

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Васильєв О. Характеристики Програмування C++ в прикладах і задачах. Навч. пос. Збільшений формат В5 Видавництво Ліра-К., 2020 382 с.
2. Івохін, Є.В.; Махно, М.Ф.; Піскунов, О.Г. Розробка додатків засобами мови програмування C#: Навч.-метод. посібник для проведення лабораторних робіт для студентів вищих навчальних закладів спеціальності «системний аналіз» /Є.В.Івохін, М.Ф.Махно, О.Г.Піскунов. – К.: Видавничо-поліграфічний центр "Київський університет", 2021. – 100 с
3. Горчинський С. Обґрунтування вибору мови програмування для початкових курсів програмування. Вісник Національного університету "Чернігівський колегіум" імені Т. Г. Шевченка Том 180 № 24 (2023) <https://doi.org/10.58407/visnik.232417>
4. Беркунський Є. Ю. Алгоритмізація та програмування мовами Kotlin, C/C++ : навчальний посібник / Є. Ю. Беркунський, А. Ю. Павленко. – Миколаїв : НУК, 2022. – 256 с.
5. Дашкевич А. Основи програмування на C++ [Електронний ресурс] : навч. посібник / О. О. Водка [та ін.] ; Нац. техн. ун-т "Харків. політехн. ін-т". – Електрон. текст. дані. – Харків, 2021. – 112 с. – URI: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/52280>.
6. Зеленський, О. С. Лисенко, В. С. Об'єктно-орієнтоване програмування на C++ : навч. посіб.

/ О. С. Зеленський, В. С. Лисенко. – Кривий Ріг : Держ. ун-т економ. і тех., 2023. - 215 с.
http://library.megu.edu.ua:8180/jspui/bitstream/123456789/4131/1/2023-%d0%9c%d0%92%20ООР_C%2b%2b.pdf

7. C/C++ language and standard libraries reference: [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/en-us/library/hh875057.aspx>.
8. C++ Tutorial: [Електронний ресурс]. – Режим доступу <https://www.w3schools.com/cpp/>.
9. C++ Language Tutorials: [Електронний ресурс]. – Режим доступу <http://www.cplusplus.com/doc/tutorial/>
10. Learn C++ programming language : [Електронний ресурс]. – Режим доступу <https://www.tutorialspoint.com/cplusplus/index.htm>.
11. Уроки програмування на C++ : [Електронний ресурс]. – Режим доступу <https://acode.com.ua/uroki-po-cpp/>.
12. Основи програмування на C ++ для початківців : [Електронний ресурс]. – Режим доступу <https://purecodecpp.com/uk/>.
13. Навчальні матеріали для вивчення основ консольного програмування мовою C++: [Електронний ресурс]. – Режим доступу https://sites.google.com/site/zsuelearning/c_plus.
14. Піцун, О. Й., Рудик, В. В. (2025). Класифікація вебсайтів за критерієм їх продуктивності на підставі ансамблів. Scientific Bulletin of UNFU, 35(4), 180-185. <https://doi.org/10.36930/40350421>