

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

ЗАТВЕРДЖУЮ:
Декан факультету комп'ютерних
інформаційних технологій
Ігор ЯКИМЕНКО

“ 30.07.2024 ” 2024 р.

ЗАТВЕРДЖУЮ:
Проректор з науково-педагогічної
роботи
Віктор ОСТРОВЕРХОВ

“ 30.07.2024 ” 2024 р.

РОБОЧА ПРОГРАМА

з дисципліни «Основи програмування»

Ступінь вищої освіти: бакалавр
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Штучний інтелект»

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

30.07.2024

Тернопіль – ЗУНУ
2024

Робоча програма складена на основі освітньо-професійної програми «Штучний інтелект» підготовки бакалавра галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки», затвердженої Вченою радою ЗУНУ (протокол №11 від 26.06.2024р.).

Робочу програму склала доцент кафедри КН, к.т.н. Наталія ПОРПЛИЦЯ

Робоча програма затверджена на засіданні кафедри комп'ютерних наук, протокол № 1 від 27 серпня 2024 р.

Завідувач кафедри

Андрій ПУКАС

Розглянуто та схвалено групою забезпечення спеціальності 122 „Комп'ютерні науки”, протокол № 1 від 30 серпня 2024 р.

Голова групи
забезпечення спеціальності,
д-р техн. наук, професор

Мирослав КОМАР

Гарант освітньо-професійної
програми "Штучний інтелект",
канд. техн. наук, доцент

Василь КОВАЛЬ

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 – 5	Галузь знань: 12 «Інформаційні технології»	Статус дисципліни: нормативна Мова навчання: українська
Кількість залікових модулів - 5	Спеціальність: 122 «Комп'ютерні науки»;	Рік підготовки: денна – 1; Семестр: денна – 1;
Кількість змістовних модулів - 4	Освітньо-професійна програма: «Штучний інтелект»	Лекції: денна – 30 год.; Лабораторні роботи: денна – 30 год.;
Загальна кількість годин - 150	Ступінь вищої освіти: бакалавр	Самостійна робота: денна – 78 год.; тренінг - 8. Індивідуальна робота: денна – 4 год.
Тижневих годин - 10 год., з них аудиторних – 4 год.		Вид підсумкового контролю – іспит

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. Курс "Основи програмування" охоплює методологічні основи алгоритмізації, методи та прийоми вивчення інформаційних систем, їх моделювання та реалізацію тощо. Названий 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 отримані 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

СК3- Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК8- 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.

2.4. Передумови для вивчення дисципліни.

- Знання з курсу «Інформатика» середньої школи.
- Навики роботи за комп'ютером.

2.5. Результати навчання

ПР5- 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.

ПР9- 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.

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 у програмах. 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. Показчики на структури. Передача по посиланню членів масивів структур. Об'єднання й операції з ними.

Тема 9. Вказівники та посилання

Загальний огляд. Розіменування вказівників. 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 та робота з строковими змінними.

Тема 10. функції

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

Тема 11. Рядки і операції з ними

Масиви символів у C++. Визначення довжини рядків. Копіювання і конкатенація рядків. Порівняння рядків. Перетворення рядків. Звертання рядків.

Пошук символів.

Тема 12. Використання стандартного класу стрічок string

Конструктори рядків. Зміна величини рядка і її розміру. Присвоювання, додавання і обмін рядків. Доступ до символів рядка. Копіювання рядків і порядків. Порівняння рядків. Операції пошуку. Вставка символів у рядок. Заміна і видалення символів з рядка. Операції вводу-виводу рядків.

Тема 13. Використання штучного інтелекту для створення програмного забезпечення

Автоматизація написання коду. Автоматичне тестування. Автоматизація ре факторингу. Розробка алгоритмів машинного навчання. оптимізація рішень.

4. СТРУКТУРА ЗАЛІКОВОГО КРЕДИТУ ДИСЦИПЛІНИ "ОСНОВИ ПРОГРАМУВАННЯ"

денна форма навчання	Кількість годин						
	Лекції	Лаб	Практ	СР	ІРС	Тренінг	Контрольні заходи
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 та робота з строковими змінними							
Тема 10. функції	4	4		6	1	2	Усне опитування/тестування
Тема 11. Рядки і операції з ними	4	4		6			Усне опитування/тестування
Тема 12. Використання стандартного класу стрічок string	2	2		6			Усне опитування/тестування
Тема 13. Використання штучного інтелекту для створення програмного забезпечення	2	2		6			Усне опитування/тестування
Разом	30	30	-	78	4	8	

5. ТЕМАТИКА ЛАБОРАТОРНИХ ЗАНЯТЬ

Лабораторне заняття №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 заняття №3

Програмування циклічних процесів

Мета: Навчитися складати алгоритми та програмувати задачі з циклічними процесами

Лабораторне заняття №4

Робота з масивами

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

Лабораторне заняття №5

Робота з даними типу "структура"

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

Лабораторне заняття №6

Використання функцій

Мета: Навчитися складати алгоритми та програмувати задачі з використанням функцій користувача.

Лабораторне заняття №7

Обробка стрічкових даних

Мета: Навчитися складати алгоритми та програмувати задачі з даними представленими у вигляді стрічок.

Лабораторне заняття №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 відповідно до обраної з переліку 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++, 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 різних програмних продуктів.

Порядок проведення:

1. Здійснити аналіз отриманого завдання.
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

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 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 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 з 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;
- інші види індивідуальних та групових завдань.

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
Оцінка за поточне оцінювання визначається, як середнє арифметичне з оцінок, отриманих під час занять	Середнє арифметичне із оцінок, отриманих за виконання практичних робіт 1-4	Оцінка за поточне оцінювання визначається, як середнє арифметичне з оцінок, отриманих під час занять	Середнє арифметичне із оцінок, отриманих за виконання практичних робіт 5-8	Визначається як єдина оцінка, отримана під час презентації проєкту, розробленого під час тренінгу	Визначається як єдина оцінка по 100-бальній шкалі за результат самостійної роботи за обраною тематикою	10 тестів по 2 бали=20 балів 2 теоретичні питання по 30 балів= 60 балів 1 практичний кейс = 20 балів

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	Комунікаційне програмне забезпечення (Internet Explorer, Google Chrome, Firefox)	1-13
4	Операційна система Windows, наявність доступу до мережі Internet	1-13
5	Персональні комп'ютери	1-13
6	Комунікаційне програмне забезпечення (Zoom) для проведення занять у режимі онлайн (за необхідності)	1-13
7	Комунікаційна навчальна платформа (Moodle) для організації дистанційного навчання (за необхідності)	1-13
8	Базове програмне забезпечення Microsoft Office	1-13
9	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

Основні:

1. Шпінталь М.Я., Порплиця Н.П. Методичні вказівки до виконання лабораторних робіт з дисципліни «Основи програмування» - Тернопіль: ФОП Шпак В.Б., 2021.- 44с. (електронна версія).
2. Шпінталь М.Я., Порплиця Н.П. Опорний конспект лекцій з дисципліни «Основи програмування» - Тернопіль: ФОП Шпак В.Б., 2021.- 40 с. (електронна версія).
3. Шпінталь М.Я., Порплиця Н.П. Методичні вказівки до виконання самостійної роботи з дисципліни «Основи програмування» - Тернопіль: ФОП Шпак В.Б., 2021.- 16с. (електронна версія).
4. Васильєв О. Характеристики Програмування C++ в прикладах і задачах. Навч. пос. Збільшений формат В5 Видавництво Ліра-К., 2020 382 с.
5. C++ Tutorial: [Електронний ресурс]. – Режим доступу <https://www.w3schools.com/cpp/>.
6. C++ Language Tutorials: [Електронний ресурс]. – Режим доступу <http://www.cplusplus.com/doc/tutorial/>
7. Learn C++ programming language : [Електронний ресурс]. – Режим доступу <https://www.tutorialspoint.com/cplusplus/index.htm>.
8. Уроки програмування на C++ : [Електронний ресурс]. – Режим доступу <https://acode.com.ua/uroki-po-cpp/>.
9. Основи програмування на C ++ для початківців : [Електронний ресурс]. – Режим доступу <https://purecodecpp.com/uk/>.

Додаткові:

10. C/C++ language and standard libraries reference: [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/en-us/library/hh875057.aspx>.
11. Навчальні матеріали для вивчення основ консольного програмування мовою C++: [Електронний ресурс]. – Режим доступу:

https://sites.google.com/site/zsuelearning/c_plus.

- 12.Знання-орієнтовані системи для ідентифікації інтервальних математичних моделей складних динамічних та статичних об'єктів: монографія / М. П. Дивак, А. М. Мельник, В. І. Манжула, І.Я, Співак, Н.П. Порплиця. - Тернопіль : ЗУНУ, 2024. - 288 с.
- 13.Madiudia, N. Porplytsya, K. Zelenetska, and R. Shevchuk, "Modeling dynamics of traffic for the e-commerce website in the process of its search engine optimization during the COVID-19 pandemic," Proceedings of the 2021 11th International Conference 'Advanced Computer Information Technologies', Deggendorf, Germany, September 15–17, 2021, pp. 61–64.
- 14.M. Dyvak, N. Porplytsya, A. Dyvak, O. Shidlovsky, D. Osadchuk, and V. Pryvrotsky, "Interval model for assessing the position of the recurrent laryngeal nerve at the site of surgery wound during thyroid surgery," Proc. of the 2021 11th International Conference on Advanced Computer Information Technologies (ACIT), 2021, pp. 117–120.
- 15.M. Dyvak, R. Pasichnyk, N. Porplytsya , V. Fronchko, V. Yakovi Vas, V. Bandžuchová, "A Method of Studying the Completeness of Interval Discrete Models Built on the Basis of Intelligent Computing," 2023 IEEE 13th International Conference on Advanced Computer Information Technologies (ACIT), Deggendorf, Germany, 2023, pp. 1–5.