

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
Факультет механіки, енергетики та інформаційних технологій
Кафедра Інформаційних технологій



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ПРОГРАМУВАННЯ

ОПП «Комп'ютерні науки»
спеціальність: 122 – «Комп'ютерні науки»
першого (бакалаврського) рівня вищої освіти

Львів 2024 р.

Робоча програма навчальної дисципліни **ПРОГРАМУВАННЯ** для здобувачів спеціальності 122 «Комп'ютерні науки», перший (бакалаврський) рівень вищої освіти.

Розробники: к.т.н., доц. Татомир А.В., ст.викладач Заплатинський Н.Б.

Робочу програму схвалено на засіданні кафедри **Інформаційних технологій** Протокол №1 від 12 серпня 2024 року.

Завідувач кафедри **Інформаційних технологій**



(підпис)

(Тригуба А.М.)

(прізвище та ініціали)

Робочу програму схвалено на засіданні методичної комісії факультету механіки, енергетики та інформаційних технологій Протокол №1 від 29 серпня 2024 року.

Голова методичної комісії факультету механіки, енергетики та інформаційних технологій.



(підпис)

(Ковалишин С.Й.)

(прізвище та ініціали)

1. Опис навчальної дисципліни

Галузь знань 12 «Інформаційні технології»
(шифр і назва)

Спеціальність 122 «Комп'ютерні науки»

Рівень вищої освіти: перший (бакалаврський)

Характеристика навчальної дисципліни:

Обов'язкова.

Кількість кредитів 8+1

Загальна кількість годин – 240+30

Індивідуальне науково-дослідне завдання Курсова робота
(назва)

Вид контролю: екзамен

Тижневих аудиторних годин, для денної форми навчання – 4 та 4

Співвідношення кількості годин аудиторних занять до загальної тривалості курсу становить (%):

для денної форми навчання – $\frac{104}{270} = \frac{39}{100}$

для заочної форми навчання – $\frac{40}{270} = \frac{15}{100}$

Мова викладання: українська

2. Програма навчальної дисципліни

Тема 1. Основи програмування.

1. Що таке комп'ютер, алгоритм, програма.
2. Що він вміє робити Перша програма на C++.
3. Етапи опрацювання програми

Тема 2. Система типів мови програмування.

1. Структура програми мовою C++, include, main(). Стандартні заголовкові файли, порядок трансляції. Система типів.
2. Логічний тип: зображення, оператори, вирази.
3. Літерний тип. Зображення літер.
4. Цілі типи: назви, граничні значення, зображення, оператори. Оператори присвоєння.
5. Дійсні типи. Перетворення типів. Математичні функції (cmath). Задача про трикутник.
6. Використання cin, cout.

Тема 3. Інструкції мови C++.

1. Послідовні та галужені алгоритми.

2. Інструкції if, if else, switch, break. Задача min(a,b,c).
3. Повторювані алгоритми. Інструкції циклу for, while, do while.
4. Оголошення в інструкціях.
5. Оператор ?:. Задача про кількість цифр числа.
6. Введення послідовностей чисел (стан потоку введення). Інструкції break, continue, goto.
7. Модифікатор const, тип перелік.

Тема 4. Оголошення та використання масивів.

1. Модифікатор const, іменовані константи, тип перелік.
2. Загальна структура оголошення імені, typedef, using.
3. Структуровані типи даних: масиви, рядки.
4. Використання масивів для зберігання/побудови послідовності значень.
5. Введення-виведення рядків. Функції для роботи з рядками (cstring).

Тема 5. Вказівники.

1. Оголошення вказівника, ініціалізація, використання.
2. Створення, використання та знищення динамічних змінних простих типів.
3. Зв'язок вказівників і масивів. Арифметика вказівників.
4. Створення та використання динамічних масивів.
5. Робота з двохвимірними масивами.
6. Побудова лінійного списку за допомогою вказівників.

Тема 6. Функції.

1. Функції в C++: прототип, визначення, виклик.
2. Модульне програмування.
3. Способи передавання аргументів у функцію.
4. Параметри-значення.
5. Параметри-вказівники.
6. Тип посилання, використання в параметрах функцій.

Тема 7. Можливості із застосуванням функцій.

1. Функції і масиви (вказівник + розмір, діапазон, робота з частиною масиву).
2. Вказівники, посилання і специфікатор const.
3. Вказівник на функцію (typedef, using, два «імені» однієї функції).
4. Функції вищих порядків (табуляція, застосування до кожного, числові методи).
5. Використання лямбда-виразів.
6. Масиви функцій (текстове меню).

Тема 8. Рядки. Строки символів.

1. Строка символів – масив
2. Функції для роботи з строками
3. Обробка рядків символів
4. Типізовані покажчики

Тема 9. Структури.

1. Оголошення структури, поля даних, конструктори.
2. Створення і використання екземплярів структури. Селектор імені.
3. Перевантаження оператора виведення структури.
4. Створення і використання масивів структур.
5. Функції для опрацювання структур.
6. Перевантаження арифметичних операторів.

Тема 10. Списки.

1. Поняття зв'язної структури даних.
2. Тип для моделювання ланок лінійного однозв'язного списку.
3. Побудова списку з послідовності значень.
4. Перебір елементів списку, виведення на консоль.

5. Вставлення значення у впорядкований список.
6. Вилучення списку з динамічної пам'яті.
7. Сортування списком. Алгоритм злиття впорядкованих послідовностей.

Тема 11. Дерева.

1. Сортування списком (стосується попередньої лекції).
2. Структура даних дерево. Означення.
3. Поняття рекурсії. Рекурсивні розв'язки, рекурсивні функції.
4. Алгоритми обходу двійкового дерева.
5. Дерева пошуку. Сортування деревом.
6. Збалансовані дерева, вставка, вилучення.

Тема 12. Потоки введення-виведення даних. Використання файлів.

1. Потоки даних.
2. Виведення в cout.
3. Введення з cin.
4. Введення-виведення файлів

Тема 13. Особливі випадки оголошення функцій. Простори імен. Класи пам'яті.

1. Перевантажені функції. Статичний поліморфізм.
2. Аргументи за замовчуванням функцій.
3. Вбудовані функції.
4. Функції зі змінною (невідомою) кількістю аргументів.
5. Статичні локальні змінні функцій.
6. Простори імен: призначення, оголошення, використання.
7. Класи пам'яті, діапазони доступу, зв'язність імен.

Тема 14. Класи.

1. Навіщо класи мови C++. Термінологія.
2. Синтаксис класу. Оголошення полів, методів. Прихована та доступна частини.
3. Розбиття визначення класу на файли. Тестування класу. Надсилання повідомлень.
4. Конструктори: за замовчуванням, створення, копіювання. Деструктор.
5. Порівняння об'єктів, використання вказівника this.
6. Статичні та динамічні об'єкти, масиви об'єктів, вказівники на об'єкт.
7. Діапазон видимості класу. Вкладені оголошення типів.

Тема 15. Опрацювання помилок, Винятки.

1. Традиційні способи опрацювання помилок.
2. Використання винятків (значення вбудованих типів).
3. Оголошення і використання класів винятків.
4. Стандартні класи винятків.
5. Розгортання стека, повторне генерування винятку.
6. Можливі проблеми, породжені опрацюванням винятку.
7. Додаткові можливості використання винятків.

Тема 16. Узагальнення в програмуванні. Шаблони.

1. Від конкретних даних до довільних, від частини програми до функції, від конкретного типу до довільного.
2. Шаблон функції: оголошення, використання, явне і неявне створення екземпляра.
3. Часткова спеціалізація, перевантаження шаблону функції.
4. Шаблон класу: оголошення, використання, явне і неявне створення.
5. Часткова спеціалізація шаблону класу.

Тема 17. Огляд бібліотеки STL.

1. Базові поняття: контейнер, ітератор, алгоритм, функтор.
2. Класифікація ітераторів, моделі ітераторів.
3. Класифікація контейнерів, спільні можливості.

Тема 18. Використання шаблонів класів і функцій.

1. Наслідування шаблонів `template<template>`; `template<class>`; `class<template>`.
2. Включення шаблонів.
3. Вкладений шаблон.
4. Шаблон – параметр шаблону.
5. Клас з шаблонним методом.
6. Шаблон і дружні функції.

Тема 19. Послідовні контейнери STL.

1. Спільні риси та специфічні можливості. Особливості внутрішнього влаштування
2. Загальні способи використання послідовних контейнерів.
3. Допоміжний «говіркий» клас. Порівняння ефективності.
4. Спеціальні операції зі списком.

Тема 20. Стек, черга, пріоритетна черга.

1. Особливості реалізації та інтерфейсу.
2. Приклади використання: обхід дерева, польський запис, моделювання обслуговування.
3. Доступ до реалізації. Експерименти з пріоритетною чергою.
4. Власна реалізація контейнерного адаптера.
5. Наслідування від стандартного контейнера для вдосконалення поведінки.

Тема 21. Алгоритми.

1. Концепція функтор та її реалізація. Об'єкт-функція.
2. Стандартні функтори, композиція функторів.
3. Каталог алгоритмів.
4. Приклади використання алгоритмів.

3. Структура навчальної дисципліни

Назви тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб.	інд.	с.р.		л	п	лаб.	інд.	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
	Семестр 2						Семестр 2					
Тема 1.	12	2		4		6	12					12
Тема 2.	10	2		4		4	10					10
Тема 3.	12	2		4		6	12	2		2		8
Тема 4.	10	2		4		4	10	2		2		6
Тема 5.	12	2		4		6	12	2		2		8
Тема 6.	10	2		4		4	10	2		2		6
Тема 7.	12	2		4		6	12			2		10
Тема 8.	12	2		4		6	12			2		10
<i>Іспит</i>	30	-	-	-	-	30	30	-	-	-	-	30
Усього годин	120	16	0	32	0	72	120	8	0	12	0	100
Індивідуальні завдання												
КР	-	-	-	-		-	-	-	-	-		-
Усього годин	120	16	0	32	0	72	120	8	0	12	0	100

	Семестр 3						Семестр 3					
Тема 9.	6	2		2		2	6			2		4
Тема 10.	6	2		2		2	6					6
Тема 11.	6	2		2		2	6	2		2		2
Тема 12.	6	2		2		2	6					6
Тема 13.	6	2		2		2	6	2		2		2
Тема 14.	6	2		2		2	6					6
Тема 15.	6	2		2		2	6					6
Тема 16.	6	2		2		2	6	2		2		2
Тема 17.	6	2		2		2	6					6
Тема 18.	8	2		2		4	8	2		2		4
Тема 19.	8	2		2		4	8					8
Тема 20.	8	2		2		4	8					8
Тема 21.	12	4		4		4	12			2		10
<i>Іспит</i>	30	-	-	-	-	30	30	-	-	-	-	30
Усього годин	120	28	0	28	0	64	120	8	0	12	0	100
Індивідуальні завдання												
<i>КР</i>	30	-	-	-	-	30	30	-	-	-	-	30
Усього годин	150	28	0	28	0	94	150	8	0	12	0	130

4. Теми лабораторних робіт

№ з/п	Назва теми	Кількість, год.
<i>Семестр 2</i>		
1	Робота в інтегрованому середовищі розробки Microsoft Visual Studio	4
2	Ввід та вивід інформації в консольному режимі. Потоківі та форматовані операції мови C++	4
3	Лінійні програми. Обчислення арифметичних виразів та математичних функцій	4
4	Оператори розгалуження. Оператор <i>if</i>	4
5	Оператори розгалуження. Оператор <i>switch</i>	4
6	Робота з елементами одновимірних масивів	4
7	Робота з елементами багатовимірних масивів	4
8	Прості цикли із відомим числом повторів	4
<i>Семестр 3</i>		
9	Прості цикли із невідомим числом повторів	2
10	Програмування з використанням функцій	4
11	Вкладені цикли та задачі обробки масивів	4
12	Використання рекурсії та аналіз її типів	4
13	Робота із файлами	2
14	Генерування псевдовипадкових чисел	4
15	Робота із рядковими та символьними змінними	4
16	Програмування з використанням структур	4

5. Теми винесені на самостійне вивчення

№ з/п	Назва теми
1	Алгоритмічна складність. Поліноміальна та неполіноміальна складність алгоритмів.
2	Програмне забезпечення ПК. Мови програмування.
3	Записи, множини, покажчики.
4	Процедури створення інтерфейсу прикладного застосування.
5	Засоби інтегрованого середовища розробки програмного коду (вбудований відладчик, оглядач проекту, сховище об'єктів та довідкова система).
6	Бібліотеки компонентів, базові класи оболонки програмного середовища.
7	Помилки програмування. Чистий код.
8	Порядок застосування виключних ситуацій, їх протоколювання. Коди помилок при виникненні виключних ситуацій.
9	Сучасні тенденції розвитку програмних середовищ.
10	Робота класів та об'єктів, полів, властивостей, методів, повідомлень та подій.
11	Розв'язання задач з масивами.
12	Розв'язання задач з рядками.
13	Розв'язання задач з даними типу структура.
14	Розв'язання задач з файлами.
15	Чистий код. Перевірка чистоти коду.
16	Переваги об'єктно-орієнтованого програмування.
17	Базові алгоритми сортування. Сортування дерев даних.
18	Системи числення та їх різновиди.
19	Особливості класів – перевантаження операторів.
20	Особливості класів – перетворення типу. Явні конструктори.
21	Особливості класів – коли необхідно оголошувати конструктор копіювання і оператор присвоєння.
22	Модульне тестування – чому тестування важливе. Як написати хороший тест.
23	Модульне тестування – як додати до рішення проект модульного тестування.
24	Модульне тестування – структура класу тестів. Методи-тести.
25	Модульне тестування – різновиди статичних методів класу Assert.
26	Узагальнення структури – від конкретних алгоритмів до загального.
27	Узагальнення структури – маскування структури за допомогою спеціального класу.
28	Узагальнення структури – налаштування загального алгоритму під конкретні потреби.

6. Індивідуальні завдання:

Тема(и) курсових робіт, завдання - .

1. Розробка прикладної програми для порівняльної оцінки методів сортування даних
2. Розробка додатку для процесів управління підприємством.....
3. Інформаційно-аналітична система підтримки прийняття рішень із узгодження складових технологічної системи
4. Розробка інформаційно-аналітичних додатків в середовищі для обробки даних.
5. Розробка інформаційних ботів для підтримки прийняття рішень в сфері

7. Методи навчання:

1. Словесні методи (розповідь, пояснення, бесіда, лекція).

2. Наочні методи:

– ілюстрація (картинки, таблиці, моделі, муляжі, малюнки тощо);

– демонстрування: навчальне відео чи його фрагменти; інтерактивні презентації, діючий код імітаційної моделі, компілювання та моделювання; експеримент, спостереження, досліди та аналіз результатів тощо.

3. Практичні методи: досліди, вправи, самостійна робота. Лабораторні та практичні роботи, розрахункові, реферати.

8. Очікувані результати навчання з дисципліни:

Очікуваними результатами навчання з дисципліни «Алгоритми та структури даних» є набуття студентами *інтегральних компетентностей* – здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

Загальних компетентностей – ЗК2 Здатність застосовувати знання у практичних ситуаціях. ЗК3 Знання та розуміння предметної області та розуміння професійної діяльності. ЗК7 Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Спеціальних компетентностей – 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. СК17 Здатність проектувати та реалізовувати програмне забезпечення для вирішення різноманітних прикладних задач інтелектуальної комп'ютеризації у сфері природокористування із використанням різних мов програмування, сучасних бібліотек крос-платформного програмування, проводити тестування на різних апаратних платформах, впроваджувати і підтримувати роботу інформаційних систем на сучасних платформах.

Програмні результати навчання – ПРН5 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. ПРН9 Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач у галузі комп'ютерних наук.

9. Методи контролю:

1. Усне опитування (фронтальне, індивідуальне детальний аналіз відповідей студентів).

2. Письмова аудиторна та поза аудиторна перевірка (побудова алгоритмів, створення програмного коду, редагування коду, розрахункові, вирішення задач і прикладів, виконання графічних матеріалів, схем, підготовка різних відповідей, рефератів, контрольні роботи тощо).

3. Практична перевірка (проведення різних вимірів, збір, систематизація та опрацювання складання, налагодження, розробка документації, виконання практичної роботи, аналіз виробничої інформацію, рішення професійних завдань, ділові ігри і т.д.

4. Стандартизований контроль (тести, контрольна робота).

Види контролю: Поточний контроль, проміжна та семестрова атестація.

10. Розподіл балів, які отримують студенти

Семестр 2

Поточне тестування та самостійна робота (разом 50 балів)									Підсумковий контроль (іспит)	Сума
T1	T2	T3	T4	T5	T6	T7	T8			100
6	6	6	6	6	6	6	8		50	

Семестр 3

Поточне тестування та самостійна робота (разом 50 балів)													Підсумковий контроль (іспит)	Сума	
T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21			100
3	4	4	4	3	4	4	4	4	4	4	4	4		50	100

T1, T2 ... T14 – теми практичних робіт.

11. Методичне забезпечення

Навчально-методичні матеріали до семінарських, практичних і лабораторних занять; підручники і навчальні посібники; індивідуальні навчально-дослідні завдання; контрольні роботи; текстові та електронні варіанти тестів для поточного і підсумкового контролю, методичні матеріали для організації самостійної роботи студентів, виконання індивідуальних завдань, курсових і дипломних робіт.

Посилання на матеріали розміщені в Мудл та в Інтернеті.

12. Рекомендована література

Базова

1. Бублик В.В. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. К.: ІТ-книга, 2015. 624 с.
2. Дудзяний І.М. Програмування мовою С++. Частина 1: Парадигма процедурного програмування: навчальний посібник / І.М. Дудзяний. Львів: ЛНУ ім. І.Франка, 2013. 468 с.
3. Каплун В. А. Технологія програмування. Лабораторний практикум : навчальний посібник / В.А. Каплун, Ю.В. Баришев, А.В. Остапенко. Вінниця: ВНТУ, 2015. 125 с.
4. Основи програмування мовою С++ / Путянін Є.П., Степанов В.П., Пчелінов В.П., Долженкова Т.Г., Матат О.О. // Харків.: «Компанія СМІТ». 2015. 319 с.
5. С++. Теорія та практика : Навч. посібник / [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.] ; за ред. О. Г. Трофименко, 2011. 587 с.
6. Ткачук В.М. Програмування на С++ : Лабораторний практикум / В.М. Ткачук. Івано-Франківськ : Видавництво Прикарпатського національного університету ім. В.Стефаника, 2011. 160 с.
7. Ярошко С. А. Методи розробки алгоритмів. Програмування мовою С++: навч. посібник / С.А. Ярошко, О.С. Ярошко. Львів: ЛНУ ім. І.Франка, 2022. 248 с. <https://lnuittutor.github.io/>

Допоміжна

8. Бондаренко М.Ф. Конспект лекцій «Алгоритмічні мови та програмування» / Бондаренко М.Ф., Бритік В.І., Свиляр М.К. // Харків.: «Компанія СМІТ». 2012. 220 с.
9. Глинський Я.М. С++ і С++ Builder / Глинський Я.М., Анохін В.С., Рязьська В.А. // Львів: Деол, СПД Глинський, 2013. 192 с.

13. Інформаційні ресурси

1. Бібліотечно-інформаційні ресурси – книжковий фонд, періодика та фонди на електронних носіях бібліотеки ЛНУП, державних органів науково-технічної інформації, наукових, науково-технічних бібліотек та інших наукових бібліотек України.

2. Електронний ресурс стосовно ІТ компаній Львівського регіону:
http://www.invest-lvivregion.com/it-компанії_ua_285cms.htm

3. Задачі програмування із прикладами розв'язку: <http://purecodecpp.com/uk/archives/433>

4. Задачі програмування із прикладами розв'язку:
<http://library.nuft.edu.ua/ebook/datathree.php?ID=138>

5. Задачі програмування із прикладами розв'язку:
<http://abramov.org.ua/blog/category/opp/obchislennya-%D1%96z-zber%D1%96gannyam-posl%D1%96dovnosti/>

6. ІТ компанії Львова:
http://it-catalogue.net/ru/component/companies_cat/companies/458/all/default/all/main.html

7. Книжки з програмування: як читати і що саме:
<https://dou.ua/lenta/articles/programming-books/>

8. Код тестових програм. URL: github.com/krenevych/algo

9. Українська технічна література. Програмування:
<https://ukrtechlibrary.wordpress.com/tag/програмування/>

10. Портал об'єктно-орієнтованого програмування: <http://oop.in.ua/tag/FAQ/>

11. Eolymp. URL: eolymp.com