

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теоретичної кібернетики

«ЗАТВЕРДЖУЮ»
Заступник декана
з навчальної роботи

Кашпур О. Ф.
« 28 » 08 2020 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ПРОГРАМУВАННЯ**

для студентів

галузь знань	12 «Інформаційні технології»
спеціальність	122 «Комп'ютерні науки»
освітній рівень	бакалавр
освітня програма	«Інформатика»

вид дисципліни **обов'язкова**

Форма навчання	денна
Навчальний рік	2020/2021
Семестр	1, 2
Кількість кредитів ECTS	10
З них семестр 1	4
семестр 2	6
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: **к.ф.-м.н, доц. Ставровський А. Б.** (лекції, лабораторні заняття),
асистенти Коваль Ю. В., Кондратюк С. С. (лабораторні заняття)

Пролонговано: на 2021/2021 н. р. « 7 » 05 2021 р.
на 20__/20__ н. р. «__»__ 20__ р.

Розробник: Ставровський Андрій Борисович, канд. фіз.-мат. н., доцент,
доцент кафедри теоретичної кібернетики

ЗАТВЕРДЖЕНО

Зав. кафедри теоретичної кібернетики


_____ (підпис)

(Крак Ю.В.)
(прізвище та ініціали)

Протокол № 1 від «27» 08 2020 р.

Схвалено гарантом освітньо-професійної програми «Інформатика»


_____ (підпис) (Омельчук Л. Л.)

«28» 08 2020 р.

Схвалено науково-методичною комісією
факультету комп'ютерних наук та кібернетики

Протокол від «28» 08 2020 року № 1

Голова науково-методичної комісії _____ (Омельчук Л. Л.)
(підпис) (прізвище та ініціали)

«28» 08 2020 року

1 Мета дисципліни – знайомство з початками програмування, базовою термінологією, засобами керування порядком обчислень у мові програмування високого рівня, класичними задачами, оволодіння елементами технології створення програм.

2 Попередні вимоги до опанування або вибору навчальної дисципліни

1. *Знати:* математику та інформатику в об'ємі шкільного курсу
2. *Вміти:* застосовувати знання з дисциплін шкільного курсу до розв'язання задач
3. *Володіти елементарними навичками:* роботи з комп'ютером

3 Анотація навчальної дисципліни

Навчальна дисципліна вивчає: зображення числових та інших даних, організація керування порядком обчислень, підпрограми, базові структури даних, класи та основні поняття ООП, робота з файлами та потоками, контейнери та ітератори, зв'язані структури даних, поняття складності алгоритмів, алгоритми сортування, робота з графами.

Дисципліна використовує поняття дискретної математики й забезпечує алгоритми для неї, є базовою для дисципліни «Об'єктно-орієнтоване програмування».

Викладається в 1-му й 2-му семестрах, обсяг 300 год. (10 кредитів ECTS), з них лекції – 68 год., лабораторні заняття – 68 год., консультації – 6 год., самостійна робота – 158 год. Семестри закінчуються іспитами.

4 Завдання (навчальні цілі)

Розвивати:

- практичні навички з розв'язування навчальних і практичних задач програмування,
- здатність вчитися й оволодівати сучасними знаннями,
- знання та розуміння предметної області та розуміння професійної діяльності,

Розвивати програмні компетентності (згідно освітньої програми):

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

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.

5 Результати навчання за дисципліною

Результат навчання (РН) (1 – знати; 2 – вміти; 3 – комунікація; 4 – автономність та відповідальність)		Форми викладання та навчання	Методи оцінювання	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
РН1.1	Знати основні поняття програмування та принципи розробки програм	Лекція, лабораторне заняття, самостійна робота	Контрольна робота (КР)	10
			Іспит	5
РН1.2	Знати базові класичні алгоритми та різновиди структур даних	Лекція, лабораторне заняття, самостійна робота	КР	10
			Іспит	5
РН2.1	Вміти проектувати, розробляти та тестувати програми	Лабораторне заняття, самостійна робота	Здача лабораторних робіт (ЛР)	25
			Іспит	15
РН3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти	Лабораторне заняття, самостійна робота	Здача ЛР	5
			Іспит	5
РН4.1		Самостійна робота	Здача ЛР	5

	Організувати свою самостійну роботу для досягнення результату		Іспит	5
РН4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Лабораторне заняття, самостійна робота	Здача ЛР	5
			Іспит	5

6 Співвідношення результатів навчання дисципліни з програмними результатами навчання

Результати навчання дисципліни	Р Н 1. 1	Р Н 1. 2	Р Н 2. 1	Р Н 3. 1	Р Н 4. 1	Р Н 4. 2
Програмні результати навчання						
<i>(з опису освітньої програми)</i>						
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 в галузі комп'ютерних наук.	+	+	+	+	+	+

7 Схема формування оцінки

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

Перший семестр

1. Контрольна робота 1: РН1.1, РН1.2 – 10 б./6 б.
2. Контрольна робота 2: РН1.1, РН1.2 – 10 б./6 б.
3. Лабораторна робота 1: РН2.1, РН3.1, РН4.1, РН4.2 – 10 б./6 б.
4. Лабораторна робота 2: РН2.1, РН3.1, РН4.1, РН4.2 – 15 б./9 б.
5. Лабораторна робота 3: РН2.1, РН3.1, РН4.1, РН4.2 – 15 б./9 б.

Другий семестр

1. Контрольна робота 3: РН1.1, РН1.2 – 10 б./6 б.
2. Контрольна робота 4: РН1.1, РН1.2 – 10 б./6 б.
3. Лабораторна робота 4: РН2.1, РН3.1, РН4.1, РН4.2 – 10 б./6 б.
4. Лабораторна робота 5: РН2.1, РН3.1, РН4.1, РН4.2 – 15 б./9 б.
5. Лабораторна робота 6: РН2.1, РН3.1, РН4.1, РН4.2 – 15 б./9 б.

Студент має право один раз перескласти контрольну роботу з можливістю отримати не більше 80% балів, призначених за роботу.

У разі низькоякісного виконання лабораторної роботи викладач має право не зарахувати лабораторну роботу або знизити за неї бали.

Типові завдання контрольних робіт

Контрольна робота № 1

Записати задане число в заданій системі числення.

Записати додатковий код заданого цілого від'ємного числа.

Описати обчислення заданого виразу.

Написати вираз-ознаку певної властивості.

Написати програму з інструкціями розгалуження для обчислень за формулою з набором умов.

Дати означення кількох базових понять програмування та мови C++.

Матеріал, що виноситься на контрольну роботу № 1: див. запитання 1-14 для підготовки до оцінювання.

Контрольна робота № 2

Написати систему рекурентних співвідношень для обчислення величини за формулою.
Написати функцію циклічного обчислення величини за рекурентними співвідношеннями.
Написати функцію введення послідовності чисел з заданою умовою закінчення.
Ілюструвати роботу програми з рекурсивною функцією.
Ілюструвати роботу програми з киданням та перехопленням винятків.
Означити кілька понять, зв'язаних з циклами, рекурсією, масивами та вказівниками.
Матеріал, що виноситься на контрольну роботу № 2: див. запитання 15-25 для підготовки до оцінювання.

Контрольна робота № 3

Написати функцію обробки числового масиву.
Ілюструвати роботу програми з обробки числового або символічного масиву.
Означити структуру або клас.
Означити структуру або клас, об'єкти яких містять вказівники на динамічні структури.
Означити кілька термінів, зв'язаних з класами, абстрактними типами даних, динамічними даними.
Матеріал, що виноситься на контрольну роботу № 3: див. запитання 35-52 для підготовки до оцінювання.

Контрольна робота № 4

Написати класи для роботи з послідовностями зв'язаних динамічних структур.
Реалізувати операції з послідовностями зв'язаних структур.
Написати абстрактний клас та ієрархію похідних класів для роботи з графами.
Реалізувати віртуальні функції в похідних класах.
Означити кілька термінів, зв'язаних з успадкуванням класів.
Матеріал, що виноситься на контрольну роботу № 4: див. запитання 53-74 для підготовки до оцінювання.

- підсумкове оцінювання (у формі іспиту в кожному семестрі):

- максимальна кількість балів які можуть бути отримані студентом: 40;
- результати навчання, які оцінюються: РН1.1, РН1.2, РН2.1, РН3.1; РН4.1, РН4.2;
- форма проведення: письмова;
- види завдань: теоретичні питання (5 по 8%, разом 40%), задачі (20% і 40%, разом 60%).

Матеріал, що виноситься на іспит у першому семестрі: див. запитання 1-34.

Матеріал, що виноситься на іспит у другому семестрі: див. запитання 35-84.

Студент допускається до іспиту, якщо в семестрі набрав не менш ніж 20 балів. Для отримання загальної позитивної оцінки з дисципліни оцінка за іспит має бути не менше ніж 24 бали.

Запитання для підготовки до оцінювання

1. Базові поняття програмування.
2. Позиційні системи числення, форми зображення цілих чисел.
3. Принципи зображення дійсних чисел.
4. Алгоритми перетворення записів чисел у різних системах.
5. Лексика мови C++, необхідні елементи програми.
6. Оголошення імен змінних, запис констант, запис арифметичних виразів.
7. Компіляція, компонування й запуск програми.
8. Бібліотечні математичні функції.
9. Запис виразів з операторами порівняння та булевими операторами.
10. Блок інструкцій.
11. Область дії оголошення імені та правила її визначення.
12. Інструкції розгалуження.
13. Інструкція вибору варіантів.
14. Семантика функції та її виклику, параметри та аргументи.
15. Цикл, умова продовження циклу.

16. Вигляд і семантика інструкції циклу while.
17. Вигляд і семантика інструкції циклу do-while.
18. Вигляд і семантика інструкції циклу for.
19. Структурованість інструкцій, переривання циклу.
20. Рекурсивні означення, рекурсивні функції.
21. Глибина рекурсії та її вплив на розміри програмного стеку.
22. Загальна кількість рекурсивних викликів та її вплив на час виконання програми.
23. Одиниця трансляції, означення та оголошення імен змінних, простори імен.
24. Особливі ситуації та винятки.
25. Перехоплення та обробка винятків.
26. Масив, елемент масиву.
27. Адреса та адресний вираз, вказівник.
28. Вказівники та індекси елементів масиву.
29. Параметри, що зображують масиви у функціях.
30. Масив, елементами якого є масиви.
31. Масиви масивів у функціях.
32. Рядки мови C, бібліотечні функції їх обробки.
33. Параметри головної функції та їх позначення.
34. Бібліотечний тип string та його переваги.
35. Поняття структури, елемент структури та його позначення.
36. Клас, об'єкт класу, атрибути й методи класу, інтерфейс класу.
37. Інкапсуляція та переваги, які вона дає.
38. Різновиди конструкторів, їх призначення та ситуації, в яких вони викликаються.
39. Деструктор: призначення та ситуації, в яких він викликається.
40. Основні частини пам'яті процесу виконання програми, вільна пам'ять.
41. Операції мови C++ створення та знищення динамічних змінних і масивів.
42. Необхідність власних конструкторів і деструкторів об'єктів із вказівниками.
43. Глибоке копіювання та його необхідність для обробки об'єктів із вказівниками.
44. Семантика переміщення та семантика копіювання.
45. Особливості конструктора переміщення.
46. Фізичний файл та потік.
47. Основні бібліотечні класи потоків «верхнього рівня».
48. Початок і закінчення роботи з потоком.
49. Стандартні потоки та файли введення та виведення.
50. Ситуації, в яких у вхідному потоці виникає помилка.
51. Особливості введення символьних, числових і рядкових даних.
52. Буферизація введення та виведення;
53. Абстрактний тип даних і його реалізація.
54. Абстрактний тип даних «стек» і структури даних для зображення стеків.
55. Абстрактний тип даних «черга» і структури даних для зображення черг.
56. Абстрактний тип даних «дек» і структури даних для зображення деків.
57. Перевантаження операторів.
58. Дружня функція.
59. Відмінності перевантаження оператора методом класу та дружньою функцією.
60. Успадкування класів, специфікатор доступу в оголошенні успадкування.
61. Сумісність за присвоюванням між похідними та базовими класами.
62. Встановлення вказівника на базовий клас на об'єкт похідного класу
63. Порядок викликів конструкторів і деструкторів для об'єкта похідного класу.
64. Віртуальний метод, синтаксичні й семантичні особливості.
65. Зв'язування виклику функції з її кодом, статичне та динамічне зв'язування.
66. Поняття поліморфізму, що дає поліморфізм.
67. Різновиди поліморфізму в C++ і засоби їх реалізації; поліморфний клас.
68. Суто віртуальна функція та її синтаксичні особливості.

69. Абстрактний клас та його головне призначення.
70. Шаблон функції та шаблонна функція.
71. Породжена функція та її створення.
72. Явне позначення типу для шаблонної функції в її виклику.
73. Шаблон класу та шаблонний клас.
74. Породжений клас та його створення.
75. Часова складність алгоритму.
76. Оцінки функції згори, знизу, точна.
77. Порядки зростання функцій, поліноміальний алгоритм, експоненційний алгоритм.
78. Бінарний пошук в упорядкованій послідовності.
79. Алгоритми сортування шляхом порівнянь та оцінки їх складності.
80. Графи та організація даних для їх зображення.
81. Способи зображення матриці суміжності.
82. Способи зображення структури суміжності.
83. Ефективність виконання типових операцій з графами залежно від їх зображення.
84. Алгоритми обходу графів та алгоритми, які їх використовують.

7.2 Організація оцінювання

Терміни проведення оцінювання в першому семестрі

1. Контрольні роботи: № 1 – до 7 тижня, № 2 – до 13 тижня.
2. Лабораторні роботи: № 1 – до 5 тижня, № 2 – до 9 тижня, № 3 – до 13 тижня.

Терміни проведення оцінювання в другому семестрі

1. Контрольні роботи: № 1 – до 8 тижня, № 2 – до 15 тижня.
2. Лабораторні роботи: № 1 – до 6 тижня, № 2 – до 12 тижня, № 3 – до 18 тижня.

Після завершення теоретичного навчання в семестрі й до офіційних перескладань іспитів контрольні роботи не проводяться й лабораторні роботи не приймаються.

7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8 Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

Перший семестр

№ п/п	Назва теми	Кількість годин		
		Лекції	Лабораторні	Самостійна робота
Частина 1. Основи організації обчислень				
1	Тема 1. Базові поняття програмування. <i>Самостійна робота:</i> знайомство з недесятковими системами числення, встановлення середовища програмування мовою С++ та початок роботи з ним.	2	2	4
2	Тема 2. Вступ до С++. Типи даних, літерали, операції, змінні. <i>Самостійна робота:</i> створення та відлагоджування простих програм мовою С++, виведення значень виразів, перетворення типів, введення значень у змінні.	2	2	4
3	Тема 3. Арифметичні операції та вирази. Бібліотечні математичні функції.	2	2	4

	<i>Самостійна робота:</i> програмування обчислень арифметичних та математичних виразів з трансцендентними функціями, знайомство з особливостями арифметичних типів.			
4	Тема 4. Порівняння та булеві операції. Інструкції розгалуження та вибору варіантів. <i>Самостійна робота:</i> запис умов, програмування обчислень, що розгалужуються.	2	2	4
5	Тема 5. Функції. Параметри та аргументи. <i>Самостійна робота:</i> виявлення підзадач у задачах, програмування параметризованих функцій, питання вибору різновидів параметрів.	2	2	4
6	Тема 6. Організація циклічних обчислень. <i>Самостійна робота:</i> визначення циклів та умов продовження в повторюваних обчисленнях, запис інструкцій циклу, використання рекурентних співвідношень для програмування обчислень елементів рекурентних послідовностей.	4	4	10
7	Тема 7. Рекурсія та рекурсивні функції. <i>Самостійна робота:</i> запис рекурсивних означень для математичних об'єктів, програмування рекурсивних функцій, визначення глибини рекурсії та загальної кількості рекурсивних викликів.	2	2	4
8	Тема 8. Особливі ситуації та винятки. <i>Самостійна робота:</i> використання винятків у задачах обробки арифметичних даних, програмування виявлення та обробки винятків.	2	2	6
	<i>Контрольна робота 1</i>			1
	Всього по частині 1	18	18	41
Частина 2. Масиви та структури				
9	Тема 1. Масиви та вказівники. <i>Самостійна робота:</i> виявлення даних, які зображуються масивами, використання масивів та вказівників у підпрограмах, використання двовимірних масивів у задачах обробки матриць.	6	6	12
10	Тема 2. Рядки мови C та їх використання. <i>Самостійна робота:</i> знайомство з бібліотечними функціями обробки рядків, їх використання для обробки послідовностей символів, обробка параметрів головної функції.	2	2	4
11	Тема 3. Структури. <i>Самостійна робота:</i> означення типів структур та їх використання в задачах.	2	2	4
	<i>Контрольна робота 2</i>			1
	Всього по частині 2	10	10	21
	ВСЬОГО за перший семестр	28	28	62

Загальний обсяг за перший семестр **120** год. (4 кредити ECTS), у тому числі:

Лекції – **28** год.

Лабораторні – **28** год.

Консультації – **2** год.

Самостійна робота – **62** год.

Другий семестр

№ п/п	Назва теми	Кількість годин		
		Лекції	Лабораторні	Самостійна робота
Частина 3. Класи, динамічні дані, успадкування класів				
12	Тема 1. Класи та об'єкти. Інкапсуляція. Конструктори та деструктор. <i>Самостійна робота:</i> означення класів для математичних та інших об'єктів, означення методів класів, використання об'єктів класів у задачах.	4	4	8
13	Тема 2. Файли та класи потоків. Робота з потоками. <i>Самостійна робота:</i> використання бібліотечних класів потоків уведення та виведення, знайомство з засобами керування введенням та виведенням.	2	2	4
14	Тема 3. Динамічні дані. Стеки та черги в масивах. <i>Самостійна робота:</i> знайомство з поняттям абстрактного типу даних, структури даних, організацією масивів та інших даних у вільній пам'яті.	4	4	12
15	Тема 4. Зв'язані динамічні структури. Зв'язані списки. <i>Самостійна робота:</i> реалізація абстрактних типів даних у класах, об'єкти яких містять вказівники на динамічні дані.	6	6	12
16	Тема 5. Успадкування та поліморфізм. <i>Самостійна робота:</i> створення ієрархій класів, означення та використання віртуальних та суто віртуальних методів.	4	4	10
17	Тема 6. Шаблони функцій та класів. <i>Самостійна робота:</i> створення шаблонів функцій, використання шаблонних функцій у програмах, створення шаблонів класів, використання шаблонних класів у програмах.	4	4	12
	<i>Контрольна робота 3</i>			1
	Всього по частині 3	24	24	59
Частина 4. Складність алгоритмів, алгоритми сортування та обробки графів				
18	Тема 1. Складність алгоритмів, оцінки складності алгоритмів. Алгоритми сортування масивів. <i>Самостійна робота:</i> реалізація алгоритмів сортування масивів, вимірювання часу роботи на масивах різних довжин, дослідження стійкості, порівняння ефективності роботи алгоритмів.	4	4	10
19	Тема 2. Графи та їх зображення. <i>Самостійна робота:</i> знайомство зі структурами даних для зображення графів, реалізація їх у класах, реалізація базових операцій обробки графів.	6	6	12
20	Тема 3. Алгоритми обходу графа, їх реалізація та використання в розв'язанні задач. <i>Самостійна робота:</i> застосування алгоритмів обходу графів до розв'язання задач обробки графів (визначення компонент зв'язності, визначення циклічності, побудова топологічного сортування тощо).	6	6	14
	<i>Контрольна робота 4</i>			1

Всього по частині 4	16	16	37
ВСЬОГО за другий семестр	40	40	96
ВСЬОГО	68	68	158

Загальний обсяг за другий семестр **180** год. (6 кредитів ECTS), у тому числі:

Лекції – **40** год.

Лабораторні – **40** год.

Консультації – **4** год.

Самостійна робота – **96** год.

Загальний обсяг **300** год. (10 кредитів ECTS), у тому числі:

Лекції – **68** год.

Лабораторні – **68** год.

Консультації – **6** год.

Самостійна робота – **158** год.

9. Рекомендовані джерела

Основні:

1. Белов Ю. А. Вступ до програмування мовою C++. Організація обчислень. / Ю.А. Белов, Т.О. Карнаух, Ю.В. Коваль, А.Б. Ставровський. – К.: ВПЦ «Київський університет», 2012. – 175 с.
2. Карнаух Т. О. Вступ до програмування мовою C++. Організація даних. / Т. О. Карнаух, Ю. В. Коваль, М. В. Потієнко, А. Б. Ставровський. – К.: ВПЦ «Київський університет», 2015. – 156 с.
3. Р.А. Веклич. Вступ до програмування мовою C++ : структури даних. / Р. А. Веклич, Т. О. Карнаух, А. Б. Ставровський. – К. : ВПЦ "Київський університет", 2018. – 99 с.
4. Страуструп Б. Язык программирования C++. Краткий курс. 2-е изд. / Страуструп Б. – М.: Вильямс, 2019. – 320 с.
5. Шилдт Г. C++: полное руководство, классическое издание. / Шилдт Г. – М.: Вильямс, 2016. – 800 с.

Додаткові:

6. Мейерс С. Эффективный и современный C++: 42 рекомендации по использованию C++11 и C++14. / Мейерс С. – М.: Вильямс, 2015. – 304 с.
7. Шилдт Г. C++: методики программирования Шилдта. / Шилдт Г. – М.: Вильямс, 2019. – 480 с.
8. Степанов А. Начала программирования. / Степанов А., Мак-Джонс П. – М.: Вильямс, 2017. – 272 с.
9. Саттер Г. Стандарты программирования на C++. / Саттер Г., Александреску А. – М.: Вильямс, 2016. – 224 с.
10. Кениг Э. Эффективное программирование на C++. / Кениг Э., Му Б. Эффективное программирование на C++. Практическое программирование на примерах. – М.: Вильямс, 2016. – 368 с.

10. Додаткові ресурси:

<https://drive.google.com/drive/folders/1FaHASCuCepCkXBLIWdBbyoXM8pteR1AI>

https://drive.google.com/drive/folders/1hjVlg1mgzEmRepuhMEIq3yJpHu6yrD_W