

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра обчислювальної математики

«ЗАТВЕРДЖУЮ»

Заступник декана

з навчальної роботи

\_\_\_\_\_ Людмила ОМЕЛЬЧУК

«\_\_» \_\_\_\_\_ 20\_\_ року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
Об'єктно-орієнтоване програмування

для студентів

галузь знань	<b>12 «Інформаційні технології»</b> <i>(шифр і назва)</i>
спеціальність	<b>124 «Системний аналіз»</b> <i>(шифр і назва спеціальності)</i>
освітній рівень	<b>бакалавр</b> <i>(молодший бакалавр, бакалавр, магістр)</i>
освітня програма	<b>«Системний Аналіз»</b> <i>(назва освітньої програми)</i>
вид дисципліни	<b>обов'язкова</b>

Форма навчання	<b>денна</b>
Навчальний рік	<b>2024/2025</b>
Семестр	<b>3</b>
Кількість кредитів ECTS	<b>3</b>
Мова викладання, навчання та оцінювання	<b>українська</b>
Форма заключного контролю	<b>іспит</b>

Викладачі:

**к.ф.-м.н., ас. Оноцький В.В.** (лекції, лабораторні заняття),

**к.ф.-м.н., ас. Колесников В.А.** (лабораторні заняття)

Пролонговано: на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.  
*(підпис, ПІБ, дата)*

на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.  
*(підпис, ПІБ, дата)*

КИЇВ – 2024

Розробники:

Клюшин Дмитро Анатолійович, д.ф.-м.н, професор кафедри обчислювальної математики,

Денисов Сергій Вікторович, асистент кафедри обчислювальної математики

Оноцький В'ячеслав Валерійович, к.ф.-м.н., асистент кафедри обчислювальної математики

ЗАТВЕРДЖЕНО

Зав. кафедри «Обчислювальної математики»

\_\_\_\_\_ (Ляшко С.І.)  
(підпис) (прізвище та ініціали)

Протокол № \_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

Схвалено. Гарант освітньо-професійної програми першого рівня вищої освіти "Системний аналіз" \_\_\_\_\_ Михайло ШАРАПОВ

Протокол від « \_\_\_\_ » \_\_\_\_\_ 20\_\_ року № \_\_\_\_

Голова науково-методичної комісії \_\_\_\_\_ Тетяна КАРНАУХ  
(підпис)

**1. Мета дисципліни** – розвиток компетенції ефективного використання ООП для створення алгоритмічно та технологічно навантажених програмних проектів. Знайомство з підходами до розробки індустріального та академічного програмного забезпечення. Покращення навичок програмування мовою C/C++.

**2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):**

1. *Знати:* синтаксис мови C++, основні поняття та концепції програмування, алгебру та математичний аналіз на базовому рівні (об'єм першого курсу університету), суть поняття алгоритму.

2. *Вміти:* створювати програми мовою C++, читати та аналізувати математичні тексти, реалізовувати прості алгоритми.

3. *Володіти елементарними навичками:* роботи з комп'ютером, пошуку інформації в інтернеті, користування системами перекладу.

**3. Анотація навчальної дисципліни:**

Навчальна дисципліна “Об’єктно-орієнтоване програмування” є складовою освітньо-професійної програми підготовки фахівців за першим (*бакалаврським*) рівнем вищої освіти. В її рамках досліджуються переваги та недоліки об’єктно-орієнтованого підходу, розглядаються підходи до ефективної розробки програмного забезпечення, практикується використання інструментів індустріальної розробки програмних систем, систем контролю версій. Також відбувається розвиток вмінь реалізації алгоритмів та вивчаються основи аналізу складності.

Дисципліна викладається у 3 семестрі 2 курсу в обсязі **90 год. (3 кредити ECTS)** зокрема: *лекції – 28 год., лабораторні – 14 год., консультації – 2 год., самостійна робота – 46 год.* У курсі передбачено **2 змістових модулі** та **1 модульна контрольна роботи**. Семестр 3 завершується **іспитом**.

В результаті вивчення навчальної дисципліни студент повинен:

**знати** поняття та підходи об’єктно-орієнтованого програмування, методи розробки індустріального програмного забезпечення, основні шаблони та антишаблони проектування, тонкощі використання мови C++.

**вміти** створювати складні програми мовою C++; застосовувати професійні інструменти розробки програм; використовувати системи контролю версій для індивідуальної та командної розробки; аргументовано обирати архітектуру, технології та алгоритми для розв’язання задач; реалізовувати самостійно потрібні варіанти алгоритмів.

Для допуску до дисципліни „Об’єктно-орієнтоване програмування” освітньо-професійної програми «Системний аналіз» студент повинен опанувати компетентності та результати навчання, які надає дисципліна „Програмування” програми «Системний аналіз».

**4. Завдання (навчальні цілі):**

Отримання компетентностей в професійній розробці програмного забезпечення для індустріальних та академічних проектів. Відповідно до освітньої кваліфікації бакалавра з системного аналізу дисципліна спрямована на досягнення таких компетентностей випускника:

- Здатність застосовувати знання у практичних ситуаціях (ЗК2)
- Здатність планувати і управляти часом (ЗК3)
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК7)
- Здатність оцінювати та забезпечувати якість виконуваних робіт. (ЗК14)

- Здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проектувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації обробки інформації, інтелектуального аналізу даних. (ФК6)
- Здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем а саме: об'єктно-орієнтований підхід при проектуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань (ФК7)
- Здатність організувати роботу з аналізу та проектування складних систем, створення відповідних інформаційних технологій та програмного забезпечення (ФК8)

### 5. Результати навчання за дисципліною:

<b>Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)</b>		<b>Форми (та/або методи і технології) викладання і навчання</b>	<b>Методи оцінювання та пороговий критерій оцінювання (за необхідності)</b>	<b>Відсоток у підсумковій оцінці з дисципліни</b>
<b>Код</b>	<b>Результат навчання</b>			
PH1.1	<i>Знати основні поняття об'єктно-орієнтованого програмування</i>	<i>Лекція, лабораторне заняття</i>	<i>контрольна робота, залік, іспит</i>	15%
PH1.2	<i>Знати основні підходи до оцінки складності алгоритмів</i>	<i>Лекція, лабораторне заняття</i>	<i>контрольна робота, іспит</i>	5%
PH1.3	<i>Знати принципи та найбільш поширені шаблони об'єктно-орієнтованого проектування програмного забезпечення</i>	<i>Лекція, лабораторне заняття</i>	<i>контрольна робота, залік, іспит</i>	5%
PH2.1	<i>Вміти створювати програми мовою C++</i>	<i>Лабораторне заняття, самостійна робота</i>	<i>захист лабораторних робіт (ЛР), контрольна робота, залік, іспит</i>	15%
PH2.2	<i>Вміти розбирати та реалізовувати алгоритми для розв'язання наближених до реальних задач</i>	<i>Лабораторне заняття, самостійна робота</i>	<i>захист ЛР</i>	20%
PH2.3	<i>Вміти використовувати інструментарій для професійної розробки програм</i>	<i>Лекція, лабораторне заняття, самостійна робота</i>	<i>захист ЛР, залік, іспит</i>	10%
PH3.1	<i>Аргументувати власний вибір підходів до розв'язання задачі, спілкуватися з колегами з питань проектування та розробки програм</i>	<i>Лабораторне заняття</i>	<i>захист ЛР</i>	10%
PH4.1	<i>Організувати свою самостійну роботу для досягнення результату</i>	<i>Самостійна робота</i>	<i>захист ЛР</i>	10%
PH4.2	<i>Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість</i>	<i>Лабораторна робота</i>	<i>захист ЛР</i>	10%

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	PH 1.1	PH 1.2	PH 1.3	PH 2.1	PH 2.2	PH 2.3	PH 3.1	PH 4.1	PH 4.2
<b>Програмні результати навчання</b>									
<i>(з опису освітньої програми)</i>									
ПР 6. Володіти сучасними методами розробки програм і програмних комплексів та прийняття оптимальних рішень щодо складу програмного забезпечення, алгоритмів процедур і операцій.	+	+	+	+		+			
ПРСАПР 2. Вміти створювати ефективні алгоритми для обчислювальних задач системного аналізу та систем підтримки прийняття рішень.		+		+	+			+	+
ПР 9. Розуміти українську та іноземну мови на рівні, достатньому для обробки фахових інформаційно-літературних джерел, професійного усного і письмового спілкування, написання текстів за фаховою тематикою.					+		+		
ПРСАПР 3. Проектувати, реалізовувати, тестувати, впроваджувати, супроводжувати, експлуатувати програмні засоби роботи з даними і знаннями в комп'ютерних системах і мережах				+	+	+	+	+	+

## 7. Схема формування оцінки.

### 7.1 Форми оцінювання студентів:

#### - семестрове оцінювання:

1. Контрольна робота: PH 1.1, PH 1.3, PH 2.1 — 15 балів/9 балів.
3. Лабораторна робота 1 (проект): PH2.1, PH3.1, PH4.1, PH4.2 – 15 балів/9 балів.
4. Лабораторна робота 2 (проект): PH2.1, PH2.3, PH3.1, PH4.1, PH4.2 – 15 балів/9 балів.
5. Лабораторна робота 3 (проект): PH1.3, PH2.1, PH2.3, PH3.1, PH4.1, PH4.2 – 15 балів/9 балів.

#### - підсумкове оцінювання у формі іспиту

- максимальна кількість балів, які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: PH1.1, PH1.2, PH1.3, PH2.1, PH2.3;
- форма проведення і види завдань: письмова із захистом відповіді, два теоретичних питання (5 та 10 балів) та два практичних завдання (по 10 балів), і ще 5 балів додаються при наявності комітів практичних завдань з інтервалом до 10 хвилин в систему контролю версій.

Студент допускається до іспиту, якщо він під час семестру набрав більше 36 балів, причому лабораторні роботи було виконано як мінімум на 60%.

Для отримання загальної позитивної оцінки з дисципліни оцінка за іспит не може бути меншою 24 балів.

### 7.2. Організація оцінювання.

#### Терміни проведення форм оцінювання в 3 семестрі:

1. Контрольна робота 1: до 8 тижня семестру.
2. Лабораторна робота 1 (проект): до 3 тижня семестру.

3. Лабораторна робота 2 (проект): до 8 тижня семестру.

4. Лабораторна робота 3 (проект): до останнього тижня семестру.

Для контрольної роботи дозволяється одне перескладання, але без поважних причин (як то хвороба) можна отримати не більше 80% балів.

Лабораторні роботи можна здавати після визначеного терміну, але за перший тиждень запізнення віднімається 20% балів, і за кожен наступний по 10% (після 4-го тижня бали далі не віднімаються, на 60% можна здати завжди).

### 7.3 Шкала відповідності оцінок

<b>Відмінно / Excellent</b>	90-100
<b>Добре / Good</b>	75-89
<b>Задовільно / Satisfactory</b>	60-74
<b>Незадовільно / Fail</b>	0-59
<b>Зараховано / Passed</b>	60-100
<b>Не зараховано / Fail</b>	0-59

## 8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять СЕМЕСТР 3

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лаборат. заняття	Сам. р-та
1.	Тема 1. Вступ у мову програмування C/C++.	2	1	4
2.	Тема 2. Основи мови програмування C++.	2	1	4
3.	Тема 3. Основи системи керування версіями Git.	2	1	4
4.	Тема 4. Функції.	2	1	4
	Контрольна робота 1			
5.	Тема 5. Показчики.	4	2	6
6.	Тема 6. ООП.	2	1	4
7.	Тема 7. Класи.	2	1	4
8.	Тема 8. Спадкування класів.	2	1	4
9.	Тема 9. Виняткові ситуації.	2	1	4
10.	Тема 10. Шаблонні функції і класи.	2	1	6
11.	Тема 11. Бібліотека STL.	2	1	4
12.	Тема 12. Контейнери.	2	1	4
13.	Тема 13. Алгоритми STL.	2	1	4
	<b>ВСЬОГО</b>	28	14	46

**Загальний обсяг 90 год.**, в тому числі:

Лекцій – **28 год.**

Лабораторні заняття - **14 год.**

Консультації – **2 год.**

Самостійна робота - **46 год.**

### Тематика лабораторних проектів:

1. Використання ООП засобів C++ для структурування набору алгоритмів (на базі генераторів псевдовипадкових чисел)

2. Реалізація класу довгої цілочисельної арифметики з ефективними алгоритмами множення

3. Реалізація класу матричних обчислень в раціональній арифметиці.

### Перелік питань на екзамен:

1. Змінні.
2. Типи даних.
3. Статична типізація та перетворення типів.
4. Константи.
5. Арифметичні вирази.
6. Умовні вирази.
7. Побітові операції.
8. Операції присвоювання.
9. Ввід і вивід у консоль.
10. Простір імен та using.

11. Умовні конструкції (if, switch).
12. Цикли.
13. Посилання.
14. Масиви.
15. Рядки.
16. Система керування версіями Git.
17. Функції (визначення, оголошення, параметри, передача аргументів за значенням та за посиланням, Константні параметри, оператор return та повернення результату).
18. Показчики.
19. Динамічні масиви.
20. Визначення класу.
21. Конструктори та ініціалізація об'єктів.
22. Оголошення та визначення функцій класу.
23. Керування доступом. Інкапсуляція.
24. Приклад класу списку для факторизації числа.
25. Дружні функції та класи.
26. Ключове слово this.
27. Статичні члени класу.
28. Структури.
29. Переліки.
30. Об'єднання (union).
31. Спадкування.
32. Віртуальні функції та їх перевизначення.
33. Абстрактні класи.
34. Деструктор.
35. Перевантаження операторів.
36. Оператори перетворення типів.
37. Виняткові ситуації.
38. Шаблонні функції і класи.
39. Контейнери vector, list, deque, set, multiset, bitmap бібліотеки STL.
40. Контейнери map, multimap, stack, queue, priority\_queue бібліотеки STL.
41. Алгоритми STL.

## 9. Рекомендовані джерела:

### Основні

1. Белов Ю.А., Карнаух Т.О., Коваль Ю.В., Ставровський А.Б. Вступ до програмування мовою C++. Організація обчислень. – К. : Видавничо-поліграфічний центр "Київський університет", 2012. – 175 с
2. Повна документація з C++ - <https://en.cppreference.com/>
3. Herbert Schildt, C++: The Complete Reference, 4th Edition. - McGraw-Hill Education. – 2003. – 1023 p.
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms 3rd Edition. MIT Press. – 2009. - 1292 p.
3. Donald E. Rnuth. Art of Computer Programming, Volume 1: Fundamental Algorithms, 3rd Edition. Volumes 1-3. Addison-Wesley Professional. - 1997

## Додаткові

1. <https://www.coursera.org/learn/analysis-of-algorithms>
2. <https://www.coursera.org/learn/algorithms-on-graphs/programming/nSER4/advanced-shortest-paths>
3. Cryptography in C and C++ /MichaelWelschenbach ; translated by David Kramer.– 2nd American ed., rev. and enl. Apress. – 2005. – 503 p.
4. Bancila M. Template Metaprogramming with C++. - 2022.