

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчально-виховної роботи
Наталія ТМЄНОВА

« 29 » серпня 2026 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ОСНОВИ ЗВОРОТНЬОЇ РОЗРОБКИ»

для здобувачів вищої освіти

галузь знань **12 Інформаційні технології**
спеціальність **121 Інженерія програмного забезпечення**
освітній рівень **бакалавр**
освітня програма **Інженерія програмного забезпечення**

вид дисципліни **вибіркова**

Форма навчання	денна
Навчальний рік	2026/2027
Семестр	7
Кількість кредитів ECTS	7
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: Іванов Є.В., доктор фізико-математичних наук, доцент

Пролонговано: на 2027/2028 н.р. _____ (підпис) _____ (ПІБ) «___» _____ 2027 р.
на 2028/2029 н.р. _____ (підпис) _____ (ПІБ) «___» _____ 2028 р.
на 2029/2030 н.р. _____ (підпис) _____ (ПІБ) «___» _____ 2029 р.

КИЇВ – 2026

Розробник: Іванов Євген В'ячеславович, д.ф.-м.н., доцент, доцент кафедри програмних систем і технологій

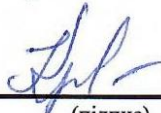
Завідувач кафедри
програмних систем і технологій


(підпис) проф. Олексій БИЧКОВ

Протокол № 18 від 17 червня 2026 року

Схвалено науково-методичною комісією факультету інформаційних технологій

Протокол № 11 від 26 червня 2026 року

Голова науково-методичної комісії 
(підпис) Ганна КРАСОВСЬКА

1. Мета дисципліни – ознайомлення студентів з методами та інструментами зворотної розробки програмного забезпечення, отримання студентами практичних умінь, необхідних для аналізу програмного забезпечення, виявлення та усунення недоліків у ньому.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

1. Знати теоретичні основи, отримані при вивченні дисциплін “Основи програмування”, “Архітектура комп’ютера”, “Основи програмної інженерії”, “Операційні системи”, “Конструювання програмного забезпечення”.
2. Вміти застосовувати практичні навички, отримані при вивченні дисциплін “Основи програмування”, “Архітектура комп’ютера”, “Основи програмної інженерії”, “Операційні системи”, “Конструювання програмного забезпечення”.

3. Анотація навчальної дисципліни:

Дисципліна надає студентам знання та вміння, необхідні для застосування методів та інструментів зворотної розробки програмного забезпечення, що є необхідними для набуття здатності забезпечувати вимоги щодо якості програмного забезпечення.

4. Завдання (навчальні цілі):

Отримання випускниками таких вмінь:

- *знати* основні методи зворотної розробки програмного забезпечення,
- *вміти* застосовувати інструменти зворотної розробки програмного забезпечення,

необхідних для забезпечення таких загальних та фахових компетентностей:

- ЗК-2: здатність застосовувати знання у практичних ситуаціях;
- ФК-6: здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки;
- ФК-11: здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Знати основні методи зворотної розробки програмного забезпечення	лекції, лабораторні заняття	контрольна робота, екзамен	50%
2.1	Вміти застосовувати інструменти зворотної розробки програмного забезпечення	лекції, лабораторні заняття	перевірка виконаних лабораторних робіт, екзамен	50%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	1.1	2.1
Програмні результати навчання		
ПР-2. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.	+	+
ПР-14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.	+	+

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

Семестрове оцінювання:

1. Контрольна робота: РН 1.1.— 30 балів.
2. Виконання практичних робіт: РН 2.1.— 30 балів.

За бажанням студентів та за погодженням з викладачем виконання лабораторних робіт може відбуватись у формі реалізації командних проєктів з тематики дисципліни. В такому випадку у якості балів за лабораторні роботи студенту зараховуються бали за роботу студента в межах командного проєкту (від 0 до 30 балів), що мають такі складові:

- до 9 балів (до 30% від 30 балів) – оцінка результату роботи команди
- до 21 балу (до 70% від 30 балів) – оцінка індивідуальної роботи студента в межах командного проєкту.

За коректні відповіді на питання викладача під час лекцій можуть бути зараховані додаткові бали, що додаються до семестрової оцінки за умови, що сума балів за контрольну роботу і лабораторні роботи (командний проєкт) та додаткових балів не перевищує 60 балів.

Підсумкове оцінювання:

- форма оцінювання – екзамен
- максимальна кількість балів які можуть бути отримані студентом – 40 балів
- результати навчання які будуть оцінюватись – РН1.1, РН2.1
- форма проведення – письмова
- види завдань – 1 теоретичне питання (24 бали) та 1 задача (16 балів).

Студент не допускається до екзамену, якщо під час семестру набрав менше ніж 36 балів.

Для отримання загальної позитивної оцінки з дисципліни оцінка за екзамен не може бути меншою 24 балів та сума балів, отриманих під час екзамену та балів, набраних під час семестру має бути не меншою за 60 балів.

7.2 Організація оцінювання:

- Контрольна робота у письмовій формі (максимум 30 балів)
- Практичні роботи (сумарно 30 балів) виконуються під час семестру та здаються на будь-якому практичному занятті під час семестру.

Рекомендований мінімум для допуску до екзамену – 36 балів.

Студент не допускається до екзамену, якщо під час семестру набрав менше ніж 36 балів.

Умова отримання позитивної підсумкової оцінки з дисципліни – отримання рекомендованого

мінімуму для допуску до екзамену, оцінка за екзамен не менше 24 балів, сума балів, отриманих під час екзамену та балів, набраних під час семестру не менше 60 балів.

7.3 Політика щодо автоматичної генерації коду

Використання інструментів автоматичної генерації коду в процесі виконання командного проєкту є дозволеним за умови, що студент

1. декларує застосування таких інструментів;
2. демонструє розуміння принципів роботи використаних інструментів, їх місця, переваг та обмежень у контексті проєкту та можливих альтернатив їх застосуванню, що впливають з раніше пройденого матеріалу даної дисципліни та попередніх дисциплін;
3. демонструє розуміння змісту згенерованих фрагментів, способів перевірки їх якості та можливих шляхів їх застосування.

7.4 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 практичних занять

№ п/п	Назва лекції	Кількість годин		
		лекції	практ.	С/Р
Частина 1 Основи зворотної розробки програмного забезпечення				
1.	Тема 1. Вступ до зворотної розробки програмного забезпечення	2	4	12
2.	Тема 2. Статичний аналіз вихідного та машинного коду	4	4	12
3.	Тема 3. Динамічний аналіз вихідного та машинного коду	4	4	14
4.	Тема 4. Статичний аналіз проміжного коду	4	4	14
5.	Тема 5. Динамічний аналіз проміжного коду	4	4	14
6.	Модульна контрольна робота	2		
Частина 2 Практика зворотної розробки програмного забезпечення				
7.	Тема 6. Приклади API поширених операційних систем	4	4	12
8.	Тема 7. Проблеми автоматизації статичного аналізу коду	4	4	12
9.	Тема 8. Проблеми автоматизації динамічного аналізу коду	4	4	12
10.	Тема 9. Зворотна розробка форматів даних	4	4	12
11.	Тема 10. Документування результатів зворотної розробки	4	4	12
	ВСЬОГО	40	40	126

Загальний обсяг 210 год., в тому числі:

Лекцій – 40 год.;

Лабораторні заняття – 40 год.;

Самостійна робота – 126 год.;

Консультації – 4 год.

9. Література:

Основна:

1. Intel 64 and IA-32 architectures software developer's manual. [Електронний ресурс]. – Режим доступу: <http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html>
2. Stallings W. Operating Systems: Internals and Design Principles (8th Edition) / W. Stallings. – Pearson. – 2014. – 800 p.
3. Зубенко В.В. Програмування. Поглиблений курс. / В.В. Зубенко, Л.Л. Омельчук – Київ: Видавничо-поліграфічний центр "Київський університет". – 2011. – 623 с.
4. SEVOCAB: Software and Systems Engineering Vocabulary [Електронний ресурс]. – Режим доступу: <http://pascal.computer.org>
5. P. Bourque, R. Fairley (eds.). SWEBOK Guide, IEEE [Електронний ресурс]. – Режим доступу: <http://www.swebok.org>
6. The Java Language Specification. [Електронний ресурс]. – Режим доступу: <https://docs.oracle.com/javase/specs/>
7. The Java Virtual Machine Specification. [Електронний ресурс]. – Режим доступу: <https://docs.oracle.com/javase/specs/>

Додаткова:

8. Cormen T. Introduction to Algorithms / T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein – MIT Press – 1990.
9. Nielson H. Semantics with applications – a formal introduction. / H. Nielson, F. Nielson. – Ser. Wiley professional computing. Wiley. – 1992.
10. Floyd R.W. Assigning meanings to programs. / R.W. Floyd // Proceedings of the American Mathematical Society Symposia on Applied Mathematics. – Vol. 19. – 1967. – pp. 19-31.