

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчально-виховної роботи
Наталія ТМЄНОВА

« 29 » *серпня* 2026 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

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	6
Кількість кредитів ECTS	5
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. _____ (_____) «____» _____ 2027 р.
(підпис) (ПІБ) (дата)
на 2028/2029 н.р. _____ (_____) «____» _____ 2028 р.
(підпис) (ПІБ) (дата)
на 2029/2030 н.р. _____ (_____) «____» _____ 2029 р.
(підпис) (ПІБ) (дата)

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 теоретичними основами, технологіями, інструментами конструювання програмного забезпечення, отримання студентами практичних умінь, необхідних для створення якісних програм за допомогою сучасних мов та середовищ програмування.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

1. Знати теоретичні основи, отримані при вивченні дисциплін “Основи програмування”, “Об’єктно-орієнтоване конструювання програм”, “Архітектура та проектування програмного забезпечення”.
2. Вміти застосовувати практичні навички, отриманні при вивченні дисциплін “Основи програмування”, “Об’єктно-орієнтоване конструювання програм”, “Архітектура та проектування програмного забезпечення”.

3. Анотація навчальної дисципліни:

Дисципліна надає студентам знання та вміння, необхідні для застосування технологій та інструментів конструювання програмного забезпечення, що є необхідними для набуття здатності розробляти програмне забезпечення різного призначення та здатності забезпечувати вимоги щодо якості програмного забезпечення.

4. Завдання (навчальні цілі):

Отримання випускниками таких вмінь:

- *знати* теоретичні основи конструювання програмного забезпечення,
- *вміти* застосовувати технології та інструменти конструювання програмного забезпечення,

необхідних для забезпечення таких загальних та фахових компетентностей:

- ЗК2: здатність застосовувати знання у практичних ситуаціях;
- ФК2: здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування;
- ФК4: здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами;
- ФК8: здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв’язання завдань інженерії програмного забезпечення;
- ФК13: здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження 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 теоретичні основи конструювання програмного забезпечення	лекції, лабораторні заняття	контрольна робота, екзамен	50%
2.1	Вміти застосовувати технології та інструменти конструювання 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			
ПРН25. Знати архітектуру програмного забезпечення та принципи проектування, володіти технологіями та методологіями розробки та реалізації програмного забезпечення та мережними технологіями, мати практичні навички з адміністрування комп'ютерних мереж в процесі розробки, розгортання та супроводження програмного забезпечення, розуміти і застосовувати програмне забезпечення мережевого обміну, обміну з сенсорами, віддаленої та розподіленої обробки інформації в інтересах інтернету речей.		+	+

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

Семестрове оцінювання:

1. Контрольна робота: РН 1.1.— 30 балів.

2. Виконання лабораторних 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, 1 завдання (максимум 30 балів)
- Лабораторні роботи (сумарно 30 балів) виконуються під час семестру та здаються на будь-якому лабораторному 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 лабораторних занять

№ п/п	Назва лекції	Кількість годин		
		лекції	лаб.	С/Р
Частина 1 Основи конструювання програмного забезпечення				
1.	Тема 1. Вступ до конструювання програмного забезпечення	2	2	6
2.	Тема 2. Мінімізація складності	4	4	8
3.	Тема 3. Питання внесення змін	4	4	8
4.	Тема 4. Конструювання для верифікації	4	4	8
5.	Тема 5. Повторне використання коду	4	6	8
6.	Модульна контрольна робота	2		
Частина 2 Практика конструювання програмного забезпечення				
7.	Тема 6. Практичні питання конструювання	2	4	8
8.	Тема 7. Технології конструювання	4	4	10
9.	Тема 8. Інструменти конструювання	4	6	10
10.	Тема 9. Застосування методів та інструментів конструювання	4	6	10
	ВСЬОГО	34	40	76

Загальний обсяг 150 год., в тому числі:

Лекцій – 34 год.;

Лабораторні заняття – 40 год.;

Самостійна робота – 76 год.

9. Рекомендовані джерела:

Основна:

1. Meyer B. Object-Oriented Software Construction, second edition. / Meyer B. – Prentice Hall. – 1997.
2. Зубенко В.В. Програмування. Поглиблений курс. / В.В. Зубенко, Л.Л. Омельчук – Київ: Видавничо-поліграфічний центр "Київський університет". – 2011. – 623 с.
3. C# Programming Guide [Електронний ресурс]. – Режим доступу: <http://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/index>
4. The Java Tutorials [Електронний ресурс]. – Режим доступу: <http://docs.oracle.com/javase/tutorial/index.html>
5. SEVOCAB: Software and Systems Engineering Vocabulary [Електронний ресурс]. – Режим доступу: <http://pascal.computer.org>

Додаткова:

1. P. Bourque, R. Fairley (eds.). SWEBOK Guide, IEEE [Електронний ресурс]. – Режим доступу: <http://www.swebok.org>
2. Unified Modeling Language Specification [Електронний ресурс]. – Режим доступу: <http://www.uml.org>
3. Gamma E. Design Patterns: Elements of Reusable Object-Oriented Software / E. Gamma, R. Helm, R. Johnson, J. Vlissides. – Addison-Wesley. – 1994. – 395 p.
4. Cormen T. Introduction to Algorithms / T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein – MIT Press – 1990.
5. Pierce B. Types and Programming Languages / B. Pierce. – MIT Press. – 2002.
6. Liskov B. A behavioral notion of subtyping / B.H. Liskov, J.M. Wing // ACM Trans. Program. Lang. Syst. – 16 (6). – 1994. – P. 1811-1841.
7. Black A.P. Object-oriented programming: Some history, and challenges for the next fifty years / A.P. Black // Information and Computation. – Vol. 231. – 2013. – P. 2-20.
8. Nielson H. Semantics with applications – a formal introduction. / H. Nielson, F. Nielson. – Ser. Wiley professional computing. Wiley. – 1992.
9. Hoare C.A.R. An axiomatic basis for computer programming / C.A.R. Hoare // Communications of the ACM. – Vol. 12 (10). – 1969. – pp. 576-580.
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.