

	СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «Серверне програмування» Рівень вищої освіти: <u>Другий (магістерський)</u> Спеціальність: <u>ЕЗ Комп'ютерні науки</u> Рік навчання: <u>2-й</u>, семестр 3-й Кількість кредитів ECTS: <u>5 кредитів</u> Назва кафедри: <u>Комп'ютерних наук та цифрової економіки</u> Мова викладання: <u>англійська</u>
Лектор курсу	к.т.н, Титарчук Євгеній Олександрович
Контактна інформація лектора (e-mail)	<u>etitarchuk@gmail.com</u>

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Навчальна дисципліна «Серверне програмування» є обов'язковою компонентою ОПП.

Загальний обсяг дисципліни 150 год.: лекції – 24 год.; практичні заняття – 18 год., самостійна робота – 108 год.

Формат проведення: лекції, практичні заняття, консультації.

Підсумковий контроль – екзамен.

При вивченні даної дисципліни використовуються знання, отримані з дисциплін: «Моделі, дані та інтелектуальні системи в AgriTech», «Теорія масового обслуговування».

Призначення навчальної дисципліни

Освітня компонента «Серверне програмування» спрямована на формування системних знань і практичних навичок проектування, реалізації, тестування та дослідження продуктивних, захищених і спостережуваних серверних систем мовою C# на платформі .NET. Наскрізним завданням дисципліни є створення високопродуктивного сервера: від проектування прикладного протоколу, асинхронного мережевого введення-виведення та керування пам'яттю до оброблення перевантаження, кешування, автентифікації, телеметрії й експериментального оцінювання продуктивності.

Під час вивчення дисципліни здобувачі опановують сокети та асинхронні API .NET, конкурентні колекції, атомарні й lock-free алгоритми, кільцеві буфери, локальне та розподілене кешування, ASP.NET Core Authentication and Authorization, OAuth 2.0, OpenID Connect, JWT і OpenTelemetry. Особлива увага

приділяється коректності багатониткового коду, обмеженню споживання ресурсів, захисту сервера від перевантаження, безпечній роботі з обліковими даними та ключами, а також вимірюванню пропускну здатності, затримки й використання ресурсів.

Мета вивчення навчальної дисципліни

Метою вивчення навчальної дисципліни «Серверне програмування» є формування професійних знань, умінь і компетентностей щодо побудови сучасних серверних систем, здатних коректно працювати за умов конкурентного доступу, змінного навантаження та обмежених обчислювальних ресурсів, а також забезпечувати належний рівень безпеки, надійності й спостережуваності.

Дисципліна спрямована на формування здатності обґрунтовувати архітектурні й алгоритмічні рішення на основі моделей, тестів та експериментальних даних; проєктувати прикладні протоколи; реалізовувати асинхронні мережеві конвеєри; керувати пам'яттю, чергами та backpressure; застосовувати кешування, автентифікацію й авторизацію; інструментувати сервер і доказово оцінювати його продуктивність.

Завдання вивчення дисципліни

Завданням навчальної дисципліни є набуття здобувачами теоретичних знань і практичних навичок щодо: проєктування бінарних і текстових протоколів поверх TCP та реалізації коректного життєвого циклу з'єднання; створення високопродуктивних асинхронних серверів із використанням сокетів, «System.IO.Pipelines», пулів буферів та засобів скасування операцій; вибору між блокуваннями, конкурентними колекціями, каналами й lock-free структурами даних залежно від вимог до коректності та продуктивності; керування обмеженими чергами, backpressure і перевантаженням на основі вимірювань та моделей теорії масового обслуговування; проєктування кешів, політик інвалідації та часової залежності даних; реалізації безпечних сценаріїв входу, автентифікації й авторизації з використанням OAuth 2.0, OpenID Connect і JWT; інструментування серверів за допомогою OpenTelemetry, профілювання, навантажувального тестування та обґрунтування оптимізацій результатами відтворених експериментів; тестування, документування, розгортання та супроводу серверного програмного забезпечення.

ПЕРЕЛІК КОМПЕТЕНТНОСТЕЙ, ЯКИХ НАБУВАЄ ЗДОБУВАЧ ПРИ ВИВЧЕНІ ДИСЦИПЛІНИ ВІДПОВІДНО ДО ОСВІТНЬОЇ ПРОГРАМИ

У результаті вивчення навчальної дисципліни здобувач повинен сформулювати такі програмні компетентності:

інтегральну компетентність (ІК):

Здатність розв'язувати задачі дослідницького та/або інноваційного характеру у сфері комп'ютерних наук.

загальні компетентності (ЗК):

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК4. Здатність спілкуватися іноземною мовою.

ЗК7. Здатність генерувати нові ідеї (креативність).

спеціальні (фахові, предметні) компетентності (СК):

СК4 Здатність збирати і аналізувати дані (включно з великими) для забезпечення якості прийняття проєктних рішень.

СК5. Здатність розробляти, описувати, аналізувати та оптимізувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення.

СК6. Здатність застосовувати існуючі і розробляти нові алгоритми розв'язування задач у галузі комп'ютерних наук.

СК7. Здатність розробляти програмне забезпечення відповідно до сформульованих вимог з урахуванням наявних ресурсів та обмежень.

СК9. Здатність розробляти та адмініструвати бази даних та знань.

СК11. Здатність ініціювати, планувати та реалізовувати процеси розробки інформаційних та комп'ютерних систем і програмного забезпечення, включно з його розробкою, аналізом, тестуванням, системною інтеграцією, впровадженням і супроводом.

ПРОГРАМНІ РЕЗУЛЬТАТИ НАВЧАННЯ ВІДПОВІДНО ДО ОСВІТНЬОЇ ПРОГРАМИ

РН1. Мати спеціалізовані концептуальні знання, що включають сучасні наукові здобутки у сфері комп'ютерних наук і є основою для оригінального мислення та проведення досліджень, критичне осмислення проблем у сфері комп'ютерних наук та на межі галузей знань.

РН9. Розробляти алгоритмічне та програмне забезпечення для аналізу даних (включно з великими).

РН10. Проектувати архітектурні рішення інформаційних та комп'ютерних систем різного призначення.

RH11. Створювати нові алгоритми розв'язування задач у сфері комп'ютерних наук, оцінювати їх ефективність та обмеження на їх застосування.

RH12. Проєктувати та супроводжувати бази даних та знань.

RH13. Оцінювати та забезпечувати якість інформаційних та комп'ютерних систем різного призначення.

RH14. Тестувати програмне забезпечення.

RH17 Виявляти та усувати проблемні ситуації в процесі експлуатації програмного забезпечення, формулювати завдання для його модифікації або реінжинірингу.

Вивчення дисципліни формує у здобувачів освіти соціальні навички (soft skills): критичне та системне мислення під час аналізу конкурентної поведінки й причин деградації продуктивності; комунікативність через обговорення архітектурних рішень і захист результатів лабораторних робіт; уміння працювати в команді під час інтеграції компонентів наскрізного серверного застосунку; відповідальність за безпеку, достовірність вимірювань і якість програмного коду; креативність у пошуку компромісів між пропускнуою здатністю, затримкою, надійністю, складністю та споживанням ресурсів; навички планування експериментів, аргументації технічних рішень і професійного представлення результатів.

ПЛАН ВИВЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№ з/п	Назви теми	Форми організації навчання та кількість годин		Самостійна робота, кількість годин
		лекційні заняття	практичні заняття	
3-й семестр				
1	Тема 1. Архітектура серверних систем на C# і .NET 10. TCP та сокети в .NET (Server-side system architecture in C# and .NET 10. TCP and sockets in .NET.)	2	-	9
2	Тема 2. Проектування прикладного протоколу та високопродуктивний асинхронний TCP-сервер. (Application protocol design and high-performance asynchronous TCP server.)	2	3	9
3	Тема 3. Ефективна робота з пам'яттю та потоками байтів. (Efficient memory and byte stream handling)	2	3	9
4	Тема 4. Модель асинхронності та пам'яті .NET: Task, ThreadPool, атомарність. (.NET Async and Memory Model: Task, ThreadPool, Atomic Operations)	2	-	9
5	Тема 5. Lock-free програмування та кільцеві буфери. (Lock-free programming and ring buffers.)	2	3	9
6	Тема 6. Конкурентні колекції .NET і критерії вибору структур даних для серверного навантаження (Concurrent .NET Collections and Criteria for Selecting Data Structures for Server Workloads)	2	-	9
7	Тема 7. Асинхронні конвеєри на основі Channel<T>, bounded channels, backpressure і політики перевантаження. (Asynchronous pipelines based on Channel<T>, bounded channels, backpressure, and overload policies.)	2	3	9
8	Тема 8. Планування обслуговування та стійкість під навантаженням (Maintenance planning and resilience under load)	2	-	9
9	Тема 9. Кешування та інвалідація кешу (Caching and cache invalidation).	2	3	9
10	Тема 10. Облікові записи, вхід і модель безпеки сервера (User accounts, login, and server security model)	2	-	9
11	Тема 11. OpenID Connect, JWT та ротація ключів (OpenID Connect, JWT,	2	-	9

	and Key Rotation)			
12	Тема 12. Спостережуваність з OpenTelemetry (Observability with OpenTelemetry)	2	3	9
Разом		24	18	108

Самостійна робота здобувача вищої освіти

Самостійна робота здобувача організовується шляхом видачі індивідуального переліку питань і практичних завдань з кожної теми, які не виносяться на аудиторне опрацювання та виконання індивідуального творчого завдання.

Самостійна робота здобувача є одним із способів активного, цілеспрямованого набуття нових для нього знань та умінь. Вона є основою його підготовки як фахівця, забезпечує набуття ним прийомів пізнавальної діяльності, інтерес до творчої роботи, здатність вирішувати наукові та практичні завдання.

Виконання здобувачем самостійної роботи передбачає, за необхідності, отримання консультацій або допомоги відповідного фахівця. Навчальний матеріал навчальної дисципліни, передбачений робочою програмою для засвоєння здобувачем у процесі самостійної роботи, виносяться на поточний і підсумковий контроль поряд з навчальним матеріалом, який опрацьовувався під час аудиторних занять. Організація самостійної роботи здобувачів передбачає: планування обсягу, змісту, завдань, форм і методів контролю самостійної роботи, розробку навчально-методичного забезпечення; виконання здобувачем запланованої самостійної роботи; контроль та оцінювання результатів, їх систематизацію, оцінювання ефективності виконання здобувачем самостійної роботи.

Індивідуальні завдання здобувач виконує самостійно під керівництвом викладача згідно з індивідуальним навчальним планом.

У випадку реалізації індивідуальної освітньої траєкторії здобувача заняття можуть проводитись за індивідуальним графіком.

Види самостійної роботи

№ п/п	Вид самостійної роботи	Години	Термін виконання	Форма та метод контролю
1	Підготовка самостійних питань з тематики дисципліни (опрацювання теоретичних основ прослуханого лекційного матеріалу)	78	щотижнево / під час заліково-екзаменаційної сесії	Усне та письмове опитування, тестування, оцінювання конспекту
2	Підготовка до лекційних та практичних занять (робота з	10	щотижнево	Усне опитування, оцінювання

	інформаційними джерелами: опрацювання першоджерел)			конспекту
3	Індивідуальні творчі практичні завдання (вирішення і письмове оформлення завдань, інших робіт графічного характеру; презентації за заданою проблемною тематикою)	10	1 раз на 2 тижні / під час заліково-екзаменаційної сесії	Спостереження за виконанням, обговорення, виступ з презентацією
4	Підготовка до контрольних робіт та тестування, виконання завдань різних видів контролю під час практичних занять (самостійне опрацювання тестів відповідно до теми практичного заняття; самостійне розв'язання практичних задач, ситуаційних вправ)	10	щотижнево / під час заліково-екзаменаційної сесії	Тестування у системі Moodle
Разом		108		

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Основні

1. Бунке О. С., Серверні Web-Технології. Навчальний посібник. Київ: КПІ ім. Ігоря Сікорського, 2023. 109 с.
2. Medjaoui M., Wilde E., Mitra R., Amundsen M. Continuous API Management: Making the Right Decisions in an Evolving Landscape. Sebastopol : O'Reilly Media, 2022. 362 p.
3. Rob Miles. C# Programming Yellow Book. "Cheese" Edition 8.1. December, 2019. 217 p.
4. Mark J. Price. C# 14 and .NET 10 – Modern Cross-Platform Development Fundamentals: Build modern websites and services with ASP.NET Core, Blazor, and EF Core using Visual Studio 2026. 10th ed. Birmingham, UK : Packt Publishing Ltd., 2025. 804 p.
5. Gregg B. Systems Performance : Enterprise and the Cloud. 2nd ed. Boston : Addison-Wesley Professional, 2020. 928 p.

Додаткові

1. Kleppmann M., Riccomini C. Designing Data-Intensive Applications. 2nd ed. Sebastopol : O'Reilly Media, 2026. 672 p.
2. Eddy W. (ed.). Transmission Control Protocol (TCP). RFC 9293. STD 7. RFC Editor, 2022. DOI: <https://doi.org/10.17487/RFC9293>.
3. Best Current Practice for OAuth 2.0 Security / T. Lodderstedt, J. Bradley, A. Labunets, D. Fett. RFC 9700. BCP 240. RFC Editor, 2025. DOI:

<https://doi.org/10.17487/RFC9700>.

4. Sheffer Y., Hardt D., Jones M. JSON Web Token Best Current Practices. RFC 8725. BCP 225. RFC Editor, 2020. DOI: <https://doi.org/10.17487/RFC8725>.

5. OpenID Connect Core 1.0 incorporating errata set 2 / N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, C. Mortimore. OpenID Foundation, 2023. URL: https://openid.net/specs/openid-connect-core-1_0.html (дата звернення: 06.09.2026).

Інформаційні ресурси

1. Use Sockets to send and receive data over TCP. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/fundamentals/networking/sockets/socket-services> (дата звернення: 06.09.2026).

2. System.IO.Pipelines. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/standard/io/pipelines> (дата звернення: 06.09.2026).

3. Channels. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/core/extensions/channels> (дата звернення: 06.09.2026).

4. Thread-safe collections. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/standard/collections/thread-safe/> (дата звернення: 06.09.2026).

5. Overview of caching in ASP.NET Core. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/aspnet/core/performance/caching/overview?view=aspnetcore-10.0> (дата звернення: 06.09.2026).

6. Overview of ASP.NET Core authentication. Microsoft Learn. URL: <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/?view=aspnetcore-10.0> (дата звернення: 06.09.2026).

7. C# Generated Code Guide. Protocol Buffers Documentation. URL: <https://protobuf.dev/reference/csharp/csharp-generated/> (дата звернення: 06.09.2026).

8. .NET. OpenTelemetry. URL: <https://opentelemetry.io/docs/languages/dotnet/> (дата звернення: 06.09.2026).

СИСТЕМА ОЦІНЮВАННЯ ТА ВИМОГИ ДО КОНТРОЛЮ ЗНАНЬ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ

У кінці семестру, здобувач вищої освіти може набрати до 60% підсумкової оцінки за виконання всіх видів робіт, що виконуються протягом семестру, до 10% за показники наукової, інноваційної, навчальної, виховної роботи та студентської активності і до 30% підсумкової оцінки – за результатами підсумкового контролю.

Розподіл балів за видами навчальної діяльності

№ п.п.	Вид навчальної діяльності	Бали
Атестація 1.		
1	Участь у дискусіях на лекційних заняттях	5
2	Участь у роботі на практичних заняттях	10
3	Виконання контрольних робіт, тестування	10
4	Підготовка самостійних питань з тематики дисципліни	5
	Всього за атестацію 1	30
Атестація 2.		
1	Участь у дискусіях на лекційних заняттях	5
2	Участь у роботі на практичних заняттях	10
3	Виконання контрольних робіт, тестування	10
4	Підготовка самостійних питань з тематики дисципліни	5
	Всього за атестацію 2	30
	Показники наукової, інноваційної, навчальної, виховної роботи та студентської активності	10
	Підсумкове тестування	30
Разом		100

У кінці семестру, здобувач вищої освіти може набрати до 60% підсумкової оцінки за виконання всіх видів робіт, що виконуються протягом семестру, до 10% за показники наукової, інноваційної, навчальної, виховної роботи та студентської активності і до 30% підсумкової оцінки – за результатами підсумкового контролю.

Якщо здобувач упродовж семестру за підсумками контрольних заходів набрав (отримав) менше половини максимальної оцінки з навчальної дисципліни (менше 35 балів), то він не допускається до екзамену. Крім того, обов'язковим при мінімальній кількості балів за підсумками контрольних заходів є виконання індивідуальної творчої роботи (презентації).

Під час виконання навчальних завдань, а також завдань поточних та підсумкових контрольних заходів не допустимо порушення академічної доброчесності. Презентації та виступи мають бути авторськими та оригінальними, інформація про результати власної навчальної (наукової, творчої) діяльності – достовірною; у разі використання ідей, розробок, тверджень, відомостей мають бути посилання на джерела інформації з дотриманням норм законодавства про авторське право і суміжні права.

Програма навчальної дисципліни передбачає врахування результатів неформальної та інформальної освіти при наявності підтверджуючих документів як окремі кредити вивчення навчальних дисциплін.

Переведення балів внутрішньої 100-бальної шкали в національну

здійснюється у відповідності до шкали.

Відповідність шкал оцінок якості засвоєння навчального матеріалу

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
75-81	C		
66-74	D	задовільно	
60-65	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни