

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ

Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії

04-04-79S

СИЛАБУС

навчальної дисципліни

SYLLABUS

Інженерія програмного забезпечення		Software Engineering
Шифр за ОП	ОК 26	Code in Degree Programme
Освітній рівень: Бакалаврський (перший)		Level of Education: Bachelor's (first)
Галузь знань Інформаційні технології	12	Field of Knowledge Information Technology
Спеціальність Комп'ютерна інженерія	123	Field of Study Computer Engineering
Освітня програма: Комп'ютерна інженерія		Degree Programme: Computer Engineering

РІВНЕ – 2025

Силабус навчальної дисципліни *Інженерія програмного забезпечення* для здобувачів вищої освіти ступеня «бакалавр», які навчаються за освітньо-професійною програмою «*Комп'ютерна інженерія*», спеціальності «*Комп'ютерна інженерія*», 123. Рівне. НУВГП. 2025. 16 стор.

ОП на сайті університету: <https://ep3.nuwm.edu.ua/30336/>

Розробник силабусу: *Бойчура Михайло Володимирович, к.т.н., доцент кафедри обчислювальної техніки*

Силабус схвалений на засіданні кафедри обчислювальної техніки

Протокол № 9 від "21" лютого 2025 року

В.о. завідувача кафедри: Сидор А.І., к.т.н.

Керівник (гарант) ОП: Сидор А.І., к.т.н.

Схвалено науково-методичною радою з якості ННІ КІТІ
Протокол № 4 від "24" лютого 2025 року

Голова науково-методичної ради з якості ННІ: Мартинюк П.М., д.т.н., професор.

Попередня версія силабусу: <https://ep3.nuwm.edu.ua/30227/>.

© Бойчура М.В., 2025
© НУВГП, 2025

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ	
Інженерія програмного забезпечення	
ЗАГАЛЬНА ІНФОРМАЦІЯ	
Ступінь вищої освіти	Бакалавр
Освітня програма	Комп'ютерна інженерія
Спеціальність	123 Комп'ютерна інженерія
Рік навчання, семестр	2-й рік, 1-й семестр, 1-й рік, 1-й семестр (для бакалаврів зі скороченим терміном навчання)
Кількість кредитів	5
Лекції:	24/4 години
Лабораторні заняття:	26/12 годин
Самостійна робота:	100/134 годин
Курсова робота:	ні
Форма навчання	денна/заочна
Форма підсумкового контролю	екзамен
Мови викладання	українська та англійська
ІНФОРМАЦІЯ ПРО РОЗРОБНИКА	

<p>Лектор</p> 	<p>Бойчуря Михайло Володимирович к.т.н., доцент кафедри обчислювальної техніки</p>
<p>Вікіситет</p>	<p>http://wiki.nuwm.edu.ua/index.php/Бойчуря Михайло Володимирович</p>
<p>ORCID</p>	<p>https://orcid.org/0000-0002-9073-4037</p>
<p>Канали комунікації</p>	<p>m.v.boichura@nuwm.edu.ua, групи у месенджерах</p>
<p>ІНФОРМАЦІЯ ПРО НАВЧАЛЬНУ ДИСЦИПЛІНУ</p>	
<p>Мета та завдання</p>	
<p><i>Відштовхуючись від результатів опитувань студентів щодо якості навчання та викладання і якості освітніх програм різних спеціальностей, існує досить великий попит на поглиблене опанування певної технології замість оглядового вивчення декількох окремих. Випускники ж звертають увагу на недостатній рівень здобутих протягом навчання м'яких навичок. Дана освітня компонента передбачає проходження повного циклу веб-розробки з використанням сучасних технологій та підходів: від створення проєкту до його розгортання на веб-хостингу. У межах навчальної дисципліни студенти також матимуть можливість оволодіти тонкощами побудови порівняно великих і складних проєктів у командах та розвинути низку м'яких навичок. В силабусі враховані результати навчання, отримані викладачем Бойчурою М.В. під час підвищення кваліфікації в ІТ-компанії SoftServe (курс .NET Advanced for Teachers, період: 04.01.2023–07.03.2023).</i></p> <p>Метою навчальної дисципліни «Інженерія програмного забезпечення» є поглиблене оволодіння студентами сучасними та актуальними підходами до веб-розробки на базі фреймворку ASP.NET Core MVC.</p> <p>Основним завданням освітньої компоненти є набуття знань, умінь та навичок (компетентностей), необхідних для проєктування та реалізація порівняно великих і складних командних веб-проєктів на базі фреймворку ASP.NET Core MVC, відповідно до кваліфікації бакалавра з комп'ютерної інженерії.</p>	
<p>Оновлення</p>	

У поточну версію силабусу, в порівнянні з попередньою, внесено компетентності, результати навчання та передумови вивчення, які відповідають новій, затвердженій у 2024 році, освітній програмі.

Додатково введено викладання підтеми API, поряд з існуючими у попередній версії силабусу: *Serialization, Sessions та Deployment*. Це дозволяє забезпечити більший зв'язок даної ОК з вибіркоким компонентом Web-програмування (ВБ 1.1). Таким чином, студенти отримують можливість розробляти веб-додатки із використанням сучасної концепції розділення на *Front-End та Back-End*.

Загалом змінено номер теми: з 10 на 7; при цьому підтеми *Sessions, Serialization та Web-site deployment* доповнені викладанням API. Тоді як підтема *Partial View* теми 10 приєднана до теми 11.

Така послідовність тем дозволить готувати більш якісних фахівців. Оновлені завдання до лабораторних робіт змінені таким чином, щоб в подальшому розроблені веб-додатки могли стати хорошою базою для написання кваліфікаційних бакалаврських робіт.

Посилання на розміщення освітнього компонента на навчальній платформі Moodle, на платформі освітніх програм та їхніх освітніх компонентів

<https://exam.nuwm.edu.ua/course/view.php?id=2720>

**Передумови вивчення
(місце освітнього компонента в структурно-логічній схемі)**

Для засвоєння даної освітньої компоненти у повній мірі необхідно отримати основні знання та навички з наступних освітніх компонент: Іноземна мова (ОК 3), Вступ до спеціальності (ОК 10) та Програмування (ОК 13).

Отримані навички, серед іншого, частково можуть використовуватись при читанні освітніх компонент: Web-програмування (ВБ 1.1) та Організація баз даних (ОК 23).

Компетентності

Z1. Здатність до абстрактного мислення, аналізу і синтезу.

Z5. Здатність спілкуватися іноземною мовою.

Z8. Здатність працювати в команді.

P2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

P3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

P7. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

P14. Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.

P15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.

Програмні результати навчання (ПРН). Результати навчання (РН)*

N5. Мати знання основ економіки та управління проектами.

N8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

N12. Вміти ефективно працювати як індивідуально, так і у складі команди.

N14. Вміти поєднувати теорію і практику, а також приймати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням загальнолюдських цінностей, суспільних, державних та виробничих інтересів.

N16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

N17. Вміти проектувати, налагоджувати та обслуговувати інтегровані інтелектуальні комп'ютерні системи для підвищення ефективності бізнес-процесів.

N21. Спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською).

N23. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

N24. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

N25. Якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики.

СТРУКТУРА ТА ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№	Тема	Опис лекції	Опис лабораторного заняття
МОДУЛЬ 1.			
Змістовий модуль 1. Основи розробки веб-сайтів на базі фреймворку ASP.NET Core MVC			
1	Вступ. Огляд шаблону ASP.NET Core MVC. 2 год. лекцій 2 год. лабораторних N14.	Огляд платформи .NET. Переваги та недоліки ASP.NET Core. Структура проєкту. Вступ до патерну MVC. Фреймворк Bootstrap.	Встановлення необхідних компонентів. Створення першого проєкту. Знайомство з файлами <code>_Layout.cshtml</code> , <code>Index.cshtml</code> , <code>HomeController.cs</code> . Класи Bootstrap для роботи з кнопками, картками та текстовими полями.
2	Патерн MVC. 2 год. лекцій 2 год. лабораторних N8, N14.	Особливості роботи з представленнями. Робота з класами. Створення моделі. Властивості. Модифікатори доступу. Створення контролера. Дії контролера. Взаємодія моделі, представлення та контролера.	Створення нового контролера. Передача чисел та рядків у представлення. Створення моделі. Передача моделі-списку у представлення та у зворотній бік.

3	Командна розробка веб-застосунків. 2 год. лекцій 2 год. лабораторних N5, N8, N12, N14, N21, N24, N25.	Особливості роботи у командах. Гра PlanningPoker. Маніфест Agile. Підходи Waterfall та Scrum. GitHub. Pull requests. Branches. Jira. Story Points. Task, Story, Bug. Графіки. Створення організації. Синхронізація GitHub та Jira.	Формування студентських команд. Командна розробка тестового проєкту. Інтеграція веб-сайтів у проєкт. Робота з GitHub Desktop. Налаштування GitHub для командної розробки. Створення вітки develop. Створення віток виду id/task/description. Формування Pull request. Робота з Jira. Налаштування Jira. Створення тестових задач. Встановлення Story Points. Створення організації. Синхронізація GitHub та Jira.
4	CRUD операції. 2 год. лекцій 2 год. лабораторних N8, N12, N14.	Типи запитів. Клас DbContext. Створення таблиць бази даних. Навігаційні властивості. Ініціалізація таблиць даними. Зчитування даних з бази даних. Додавання рядка у базу даних. Оновлення даних у рядку бази даних. Видалення рядка бази даних. Мова LINQ. Фреймворк Entity Framework. Інструмент Migration.	Microsoft SQL Server 2019. Створення схеми даних (3 пов'язані таблиці). Встановлення навігаційних властивостей. Міграція. Ініціалізація таблиць даними. Додавання, зчитування, редагування та видалення даних. Фільтрація даних. Представлення скетчів проєктів. Вибір TeamLeader'a.
5	Пошарова архітектура. Сервіси. 2 год. лекцій 2 год. лабораторних N8, N12, N14, N16, N23.	Відділення роботи з базою даних від UI. Відділення бізнес-логіки від UI. Створення сервісу. Дизайн-патерн Dependency Injection. Взаємодія декількох проєктів.	Розділення єдиного проєкту на 3: BusinessLogic, DataAccess, UI. Створення сервісів у BusinessLogic. Додавання сервісів засобами Dependency Injection. Встановлення необхідних пакетів. Представлення ступеня виконання проєкту.
МОДУЛЬ 2.			
Змістовий модуль 2. Сучасні технології та підходи до розробки веб-сайтів на базі фреймворку ASP.NET Core MVC			
6	Asynchronous programming. Roles. 2 год. лекцій 2 год. лабораторних N8, N12, N14.	Стандартні та користувацькі асинхронні методи. Метод Task.Run(). Система Identity. Створення ролей. Призначення ролей. Ініціалізація ролей. Методи розширення. Ключове слово static. Атрибут Authorize.	Перехід від синхронності до асинхронності. Створення Story в Jira. Введення ролей. Призначення ролей. Розподіл функціоналу між ролями. Представлення ступеня виконання проєкту.
7	API. Sessions. Deployment. 2 год. лекцій 2 год. лабораторних N5, N8, N12, N14, N16, N17, N23.	Поняття API. Коди відповідей. Серіалізація. Політика CORS. Випадки використання сесій. Перенесення бази даних у глобальну мережу. Розгортання веб-сайту. Оптимізація розміру додатку.	Розробка API. Маршрутизація. Застосування серіалізації. Налаштування політики CORS. Інтеграція сесій у веб-сайт. Використання OpenAPI API. Перенесення бази даних у глобальну мережу. Розгортання веб-сайту. Оптимізація розміру додатку. Представлення ступеня виконання проєкту.

8	Scaffolded items. 2 год. лекцій 2 год. лабораторних N14.	Система Identity. Адаптація Scaffolded items під потреби проєкту. Архітектура Razor Pages	Додавання тих Scaffolded items, які потребуються в межах проєкту. Редагування Razor Pages під власні потреби. Налаштування параметрів автентифікації. Пов'язування Identity з електронною поштою. Представлення ступеня виконання проєкту.
9	Repository Pattern. Clean Architecture. 2 год. лекцій 2 год. лабораторних N5, N8, N12, N14, N16, N23, N24.	Суть патерну Repository. Інтеграція патерну Repository. Generic Types. Суть Clean Architecture. Інтеграція Clean Architecture у проєкт. Підключення необхідних пакетів. Модифікатори доступу.	Інтеграція патерну Repository у проєкт DataAccess. Generic Types. Асинхронність. Перейменування проєктів у відповідності до концепції Clean Architecture. Налаштування зв'язків та модифікаторів доступу. Підключення необхідних пакетів. Налаштування Dependency Injection. Представлення ступеня виконання проєкту.
10	Specifications. DTO. Mapping. 2 год. лекцій 2 год. лабораторних N8, N12, N14, N16, N23, N24.	Випадки використання специфікацій. Lambda-вирази. Data Transfer Objects. Mapping об'єктів та списків об'єктів.	Інтеграція специфікацій у проєкт. Формування складних запитів до бази даних. Обмін даними між проєктами на основі DTO. Mapping між моделями та їх DTO-аналогами. Mapping у випадку списків. Представлення ступеня виконання проєкту.
11	Validation. Pagination. Partial View. 2 год. лекцій 2 год. лабораторних N8, N14, N16, N23.	Валідація на стороні сервера. Валідація на стороні клієнта. Атрибути валідації. Користувацькі атрибути. Tag-хелпери. Реалізація пагінації. Структуризація засобами часткових представлень.	Забезпечення коректної валідації усіх даних. Введення пагінації даних. Використання часткових представлень. Представлення ступеня виконання проєкту.
12	Підведення підсумків та подальші перспективи. 2 год. лекцій 2 год. лабораторних N5, N8, N12, N16, N21, N23, N24, N25.	Етапи проведення співбесід у контексті фреймворку ASP.NET Core MVC. Формування резюме на основі пройденого матеріалу з даної дисципліни. Організаційні моменти представлення проєктів. Принципи SOLID, Big Design Up Front, DRY, KISS тощо. Рекомендації Coding Conventions та відлагодження.	Узагальнення набутих навичок у вигляді резюме. Проведення тренувальних співбесід. Представлення ступеня виконання проєкту.
13	Захист проєктів. 2 год. лабораторних N5, N8, N12, N16, N21, N23, N25.	–	Командне представлення проєктів.

Форми, методи та технології навчання

Форми навчання	<ul style="list-style-type: none"> • очна (денна), можливо з елементами дистанційного навчання; • заочна.
Форми навчального процесу	<ul style="list-style-type: none"> • навчальні заняття (лекції, лабораторні заняття, консультації); • самостійна робота здобувачів; • робота в наукових бібліотеках та мережі Інтернет; • контрольні заходи (поточна складова оцінювання, модульні контролі, підсумковий контроль).
Методи та технології навчання	<ul style="list-style-type: none"> • робота в малих групах (команді) та індивідуальна робота; • проєктна технологія; • аналіз конкретних ситуацій (case study): ситуація-оцінка; • контекстне навчання; • проблемне навчання.
Процес навчання включає, зокрема, наступне	<ul style="list-style-type: none"> • написання комп'ютерних програм; • відлагодження програм; • Code Review; • слідування рекомендаціям Coding Conventions та принципу Big Design Up Front.
Засоби навчання	<ul style="list-style-type: none"> • відеозапис лекцій; • презентація; • методичні вказівки; • конспект лекцій; • різні тьюторіали.

Інструменти, обладнання, програмне забезпечення

- середовище розробки: Microsoft Visual Studio 2022 (з компонентами для розробки веб-застосунків із застосуванням ASP.NET та SQL Server);
- Git термінал: GitHub Desktop;
- система керування проєктами Jira;
- веб-хостинг somee.com.

Порядок оцінювання програмних результатів навчання/результатів навчання

Студент може отримати 100 балів, враховуючи наступну розбаловку:

- 1) модульні контролі: 40 балів;
- 2) поточний контроль: 26-36 бали;
- 3) командне довгострокове завдання: 24 балів;
- 4) додаткові бали: 0-10 балів.

Розподіл балів:

- 1) за модульні контрольні роботи:
 - модульний контроль №1 (20 балів):
 - Рівень 1 – 19 запитань по 0.5 балів за кожне.
 - Рівень 2 – 6 запитань по 0.9 балів за кожне.
 - Рівень 3 – 3 запитання по 1.7 балів за кожне.

• модульний контроль №2 (20 балів):

Рівень 1 – 19 запитань по 0.5 балів за кожне.

Рівень 2 – 6 запитань по 0.9 балів за кожне.

Рівень 3 – 3 запитання по 1.7 балів за кожне.

2) за поточний контроль (26-36 бали):

Передбачено по 3 бали за перші 12 лабораторних робіт; у випадку правильного виконання відповідних завдань оцінка лінійно залежить від відсотка розуміння коду. За певних умов можливе повне перенесення Story Points, вказаних у Jira, в електронний журнал. У якості альтернативи студенти можуть виконувати командні проєкти з використанням інших мов чи технологій (відмінних від ASP.NET Core MVC), але за умови попереднього узгодження деталей з викладачем.

3) командне довгострокове завдання (24 балів):

На останньому лабораторному занятті передбачається захист виконаних командних проєктів у форматі відкритої пари. Для отримання максимальної оцінки студенти повинні якісно підготуватись до представлення проєкту, реалізувати низку актуального функціоналу (патернів, технологій, архітектурних підходів тощо), показати наповненість Jira і GitHub, розподілити між собою ролі при виступі, обґрунтувати актуальність проєкту, використовувати англійську мову, у випадку дистанційного представлення повинні включати відеокамеру; оцінюється також ступінь завершеності проєкту, командна робота тощо. Викладач кожному члену команди ставитиме запитання щодо розуміння програмного коду.

4) додаткові бали за вагому громадську та студентську активності (0-10 балів):

Виставляється до 10 балів за волонтерство, олімпіади, спартакіади, конкурси, конференції, написання статей, активну студентську діяльність, конкретні пропозиції з удосконалення змісту освітньої компоненти тощо.

Основні критерії, що характеризують рівень компетентності здобувача вищої освіти при оцінюванні результатів поточного та підсумкового контролів з навчальної дисципліни:

• виконання всіх видів навчальної роботи, що передбачені силабусом навчальної дисципліни;

• глибина і характер знань навчального матеріалу за змістом навчальної дисципліни, що міститься в основних та додаткових рекомендованих літературних джерелах;

• вміння аналізувати явища, що вивчаються, у їх взаємозв'язку і розвитку;

• характер відповідей на поставлені запитання (чіткість, лаконічність, логічність, послідовність тощо);

• вміння застосовувати теоретичні положення під час розв'язання практичних задач;

• вміння аналізувати достовірність одержаних результатів.

Критерії оцінювання лабораторних робіт (у % від кількості балів, виділених на завдання із заокругленням до цілого числа):

• 0% – завдання не виконано або студент не може пояснити запрограмований код;

• 40% – завдання виконано частково та містить суттєві помилки методичного або розрахункового характеру, порушені вимоги до оформлення; студент здатний пояснити запрограмований код;

- 60% – завдання виконано повністю, але містить суттєві помилки у розрахунках або в методиці, порушені вимоги до оформлення; студент здатний пояснити запрограмований код;
- 80% – завдання виконано повністю, проте містить окремі несуттєві недоліки; студент здатний пояснити запрограмований код;
- 100% – завдання виконано правильно і без зауважень; студент здатний пояснити запрограмований код.

Поєднання навчання та досліджень

Кожен студент навіть для отримання 60 балів повинен розробити досить серйозний та великий проєкт. Передбачається, що він потребуватиме попереднього аналізу предметної області, дослідження актуальності, постановки завдань дослідження, формулювання об'єкта та предмета, розробки алгоритму тощо. Таким чином, розроблені студентами проєкти можуть бути придатними для публікування результатів.

Вивчення даної освітньої компоненти може бути хорошим фундаментом для написання кваліфікаційної бакалаврської роботи.

Рекомендована література (основна, допоміжна)

Основна література

1. Lock A. *ASP.NET Core in Action*. 3rd ed. Shelter Island: Manning, 2023. 984 p.
2. Freeman A. *Pro ASP.NET Core 7*. 10th ed. Shelter Island: Manning, 2023. 1256 p.
3. Danylko J.R. *ASP.NET 8 Best Practices: Explore techniques, patterns, and practices to develop effective large-scale .NET web apps*. Birmingham: Packt Publishing, 2023. 256 p.
4. Smith J.P. *Entity Framework Core in Action*. 2nd ed. Shelter Island: Manning, 2021. 624 p.
5. Мартін Р. Чистий Agile. Назад до основ. Харків: Фабула, 2021. 224 с.
6. Омельчук Л.Л., Русіна Н.Г. Інструментальні середовища та технології програмування. Лабораторний практикум. Навчальний посібник Одеса: Айс Принт, 2020. 175 с.
7. Мартін Р. Чистий код. Створення, аналіз і рефакторинг. Харків: Фабула, 2019. 368 с.
8. Troelsen A., Japikse P. *Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming*. 11th ed. New York: Apress, 2022. 1705 p.
9. Richter J. *CLR via C#*. 4th ed. Redmond: Microsoft Press, 2012. 1594 p.

Допоміжна література

1. Коноваленко І.В. Платформа .NET та мова програмування C# 8.0: навчальний посібник. Тернопіль: ФОП Паляниця В. А., 2020. 320 с.
2. Freeman A. *Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages*. 9th ed. New York: Apress, 2022. 1286 p.
3. Marcotte C.-H. *An Atypical ASP.NET Core 6 Design Patterns Guide*. 2nd ed. Birmingham: Packt Publishing, 2022. 678 p.
4. Shellman M., Afyouni H., Pratt P.J., Last M.Z. *A Guide to SQL*. 10th ed. Kentucky: Cengage Learning, 2020. 320 p.
5. Bootstrap · The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/> (Last accessed: 18.11.2023).
6. ASP.NET Core | Open-source web framework for .NET. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (Last accessed: 18.11.2023).
7. Introduction to Identity on ASP.NET Core | Microsoft Learn. URL: <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-8.0&tabs=visual-studio> (Last accessed: 18.11.2023).
8. Manifesto for Agile Software Development. URL: <https://agilemanifesto.org/> (Last access: 14.01.2023).

Інформаційні ресурси в Інтернеті

1. <https://www.youtube.com/@mvboichura> (YouTube канал викладача з даної дисципліни)
2. <https://www.youtube.com/watch?v=phyV-OQNeRM&list=PLLWMQd6PeGY1C3YLCyAUIDJrPS9S3WdLK&index=11> (Introduction to ASP.NET MVC in C#)
3. <https://www.youtube.com/watch?v=h7alzCkmb18&list=PLLWMQd6PeGY2GVsQZ-u3DPXqwwKW8MkiP> (An Introduction To The C# Learning Cycle)
4. <https://www.youtube.com/watch?v=lkmvnjypENw> (Clean Architecture with ASP.NET Core 6)
5. <https://www.youtube.com/playlist?list=PLqWD37Mj2te0xiDW-Q58ZIG7XZry-1-bk> (Jira Tools)

ПОЛІТИКИ ВИКЛАДАННЯ ТА НАВЧАННЯ

Перелік соціальних, «м'яких» навичок (soft skills)

<i>Вміння комунікувати.</i>	<ul style="list-style-type: none">• здатність спілкуватися державною мовою як усно, так і письмово;• вміння спілкуватись та писати із використанням англomовної професійної термінології;• навички усного спілкування;• навички письмового спілкування;• вміння писати зрозумілий код.
<i>Вміння сумісно працювати.</i>	<ul style="list-style-type: none">• навички управління проектами;• здатність планувати свій час у сенсі співставлення вимог, власних знань, здібностей і дедлайнів;• здатність працювати в команді;• навички міжособистісних відносин;• вміння надавати рекомендації іншим у коректній формі.
<i>Здатність до аналізу та синтезу.</i>	<ul style="list-style-type: none">• вміння критично мислити;• здатність знаходити вихід із складних ситуацій;• здатність до навчання;• вміння комплексного вирішення проблем.
<i>Здатність застосовувати знання у практичних ситуаціях.</i>	

Дедлайни та перескладання

Дедлайн здачі лабораторних робіт – останній четвер перед початком сесії. Здача лабораторних робіт відбувається на парі або під час консультації, дата та час якої гнучко узгоджується між студентом та викладачем.

На здачу кожного з модульних контролів студенту надається одна спроба. Перший модуль здається на будь-якій лекції у жовтні, а другий – на передостанній чи останній парі згідно розкладу. Перездача окремого модульного контролю передбачена лише за виключних обставин.

У випадку, якщо студент здав обидва модулі та набрав сумарно хоча б 60 балів, то він має право отримати дану підсумкову оцінку «автоматом»; в усіх інших випадках студент повинен йти на іспит.

У разі, якщо по закінченню сесії здобувач не набрав 60 балів, його відправляють на комісію для розгляду результатів підсумкового контролю, яка рекомендує здобувачу із академічною заборгованістю покращити результати підсумкового контролю одним із наступних шляхів: повторного складання підсумкового контролю, повторного вивчення освітньої компоненти чи повторного курсу навчання.

Неформальна та інформальна освіта

Здобувачі мають право на часткове або повне перезарахування дисципліни за умови написання ними відповідної заяви, заповнення декларації та надання документів, які підтверджують ті результати навчання, які здобувач отримав (див. положення <https://ep3.nuwm.edu.ua/28363/>). Зокрема студенти можуть самостійно проходити онлайн-курси на таких навчальних платформах, як Prometheus, Coursera, edX, edEra, FutureLearn та інших, для наступного перезарахування результатів навчання. Проте доцільно попередньо узгодити з викладачем відповідність обраного онлайн-курсу сумі навчальної дисципліни. Деякий перелік підходящих курсів наведено нижче:

- Coursera – Backend Development using ASP.Net (Розробка бекенду за допомогою ASP.Net);
- Class Central – Learn ASP.NET Core MVC (.NET 6) - Full Course (Вивчайте ASP.NET Core MVC (.NET 6) - повний курс);
- Coursera – ASP.NET for Experienced Developers Specialization (Спеціалізація ASP.NET для досвідчених розробників).

Зручний пошук курсів доступний тут: <https://www.classcentral.com/>.

Окрім того, якщо з'являються обставини для здобуття неформальної чи інформальної освіти від викладачів-практиків, то пропонуються ці можливості для студентів; рекомендуються відео-уроки практикуючих програмістів з YouTube тощо.

Правила академічної доброчесності

Задля запобігання академічної недоброчесності вимагається наступне:

- кожна студентська команда повинна виконувати власний проєкт;
- кожне завдання у проєкті студент виконує самостійно, відмічаючи це у Jira та GitHub;
- студент отримує хоча б якусь оцінку лише за умови розуміння програмного коду виконаного завдання;
- студентам забороняється: плагіятити, самоплагіятити, фабрикувати, фальсифікувати, списувати, обманювати або будь-яким чином намагатись вплинути на викладача.

Залежно від виду та ступеня порушення викладач може накладати наступні санкції:

- усне або письмове зауваження від викладача;
- попередження про можливість притягнення до академічної відповідальності;
- зниження чи анулювання результатів оцінювання навчального завдання здобувача вищої освіти;
- повторне виконання навчального завдання;
- виконання іншого навчального завдання;
- призначення додаткового навчання з питань академічної доброчесності;
- призначення додаткових контрольних заходів (додаткові індивідуальні навчальні завдання, тести тощо);
- подання клопотання на ім'я ректора з метою порушення формальної процедури розгляду питання про притягнення студента до відповідальності.

За списування під час проведення модульного контролю чи підсумкового контролю студент позбавляється подальшого права здавати матеріал і у нього виникає академічна заборгованість.

Документи стосовно академічної доброчесності (про плагіат, порядок здачі курсових робіт, кодекс честі студентів, документи Національного агентства стосовно доброчесності) наведені на сторінці «Якість освіти» сайту НУВГП – <http://nuwm.edu.ua/sp/akademichna-dobrochesnistj>.

Вимоги до відвідування

Санкції за пропуски пар не передбачені. Студент має право самостійно вивчити необхідний для здачі модульних контролів та лабораторних робіт матеріал, який в повному обсязі викладено на платформі Moodle. Також викладач розміщує відеозаписи пар на YouTube. У разі необхідності проведення консультації – викладач йде назустріч.

Відвідування пари допускається із використанням власного ноутбука. Студенти не повинні порушувати дисципліну на парі.

Для студентів, які знаходяться на індивідуальному плані навчання, надаються індивідуальні завдання.



документ підписаний КЕП
Номер документа СИЛ №769
Підписувач Сорока Валерій Степанович
Підписувач (дані КЕП):
Сертифікат 3FAA9288358EC003040000009B6C3700C8C2C100