



Course of study (code) / Назва дисципліни (шифр)	Web-програмування на мові JavaScript IT1223BWPJS
Academic year / Навчальний рік - Семестр	2022/2023 –1,2 семестр
Course of study / Назва спеціальності	121 Інженерія програмного забезпечення
Educational program / Освітня програма Education - ECTS / Рівень – Кредити Status / Статус Learning language / Мова навчання	«Інженерія програмного забезпечення» Другий (магістерський) рівень -6 ECTS Обов'язкова Українська
Author / Укладач	Дмитро Медведєв, кандидат технічних наук, Державний університет економіки і технологій, e-mail: medvediev_dg@kneu.dp.ua, <a href="http://orcid.org/0000-0002-3747-1717">http://orcid.org/0000-0002-3747-1717</a> моб. +380688535681
Консультації	чт, 14.00-15.00

#### A. OBJECTIVE OF THE SUBJECT / МЕТА ТА ЗАВДАННЯ ДИСЦИПЛІНИ

Метою вивчення даної дисципліни є надання студентам знань про Web програмування, засвоєння можливостей використання JavaScript для програмування динамічних Web-сайтів і Web-інтерфейсів доступу до баз даних. Вивчення курсу вимагає цілеспрямованої роботи над вивченням спеціальної літератури, активної роботи на лекціях та практичних заняттях, самостійної роботи та виконання індивідуальних завдань. Завданнями дисципліни є набуття студентами знань щодо сучасних підходів до побудови клієнт-серверних веб-сайтів та їхньої взаємодії з базами даних

#### B. SUBJECT PROGRAM / ПРОГРАМА ДИСЦИПЛІНИ

##### Тема 1. Основи JavaScript.

Структура коду, Суворий режим - "use strict", Змінні, Типи даних, Взаємодія: alert, prompt, confirm, Перетворення типів, Базові оператори, математика, Оператори порівняння, Умовне розгалуження: if, '?', Логічні оператори, Оператор нульового злиття (??), Цикли while та for, Конструкція "switch", Функції, Function Expression, Стрілкові функції

##### Тема 2. Основи об'єктів.

Об'єкти, Копіювання об'єктів та посилання, Складання сміття, Методи об'єкта, "this", Конструктор, оператор "new", Опціональний ланцюжок '?.', Тип даних Symbol, Перетворення об'єктів у примітиви.

##### Тема 3. Типи даних.

Методи примітивів, Числа,Рядки, Масиви, Методи масивів, Об'єкти, що перебираються, Map та Set, WeakMap та WeakSet, Object.keys, values, entries, Деструктурируюче присвоєння, дата та час, Формат JSON, метод toJSON

##### Тема 4. Функції.

Рекурсія та стек, Залишкові параметри та оператор розширення, Замикання, Застаріле ключове слово "var", Глобальний об'єкт, Об'єкт функції, NFE, Синтаксис "New Function", Планування: setTimeout та setInterval, Декоратори та переадресація виклику, call/apply, Прив'язка контексту до функції, Повторюємо стрілочні функції

##### Тема 5. Класи.

Клас: базовий синтаксис, Наслідування класів, Статичні властивості та методи, Приватні та захищені методи та властивості, Розширення вбудованих класів, Перевірка класу: "instanceof", Домішки.

##### Тема 6. Проміси, async/await.

Колбеки, Промисли, Ланцюжок промісів, Промисли: обробка помилок, Promise API, Промісифікація, Мікрозавдання, Async/await

##### Тема 7. Vue.js, вступ.

Встановлення, Вступ, Що таке Vue.js?, Початок роботи, Декларативне малювання, Умови та цикли, Робота з введенням користувача, Розбиття програми на компоненти, Ставлення до елементів користувача Web Components, Примірник Vue, Синтаксис шаблонів, Обчислювані властивості та стеження, Робота з класами та стилями, Умовне відмальовування, Відображення списків, Обробка подій, Робота з формами,

##### Тема 8. Vue.js, компоненти.

Основи компонентів, Продвинуті компоненти, Реєстрація компонентів, Вхідні параметри, Події користувача, Слоти, Динамічні та асинхронні компоненти, Обробка крайніх випадків, Переходи та анімації, Анімація списків та появи/зникнення, Анімація переходів між станами,

##### Тема 9. Vue.js. Перевикористання та композиція

Домішки, Директиви користувача, Render-функції та JSX, Плагіни, Фільтри, Інструментарій, Однофайлові компоненти, Тестування, Підтримка TypeScript, Публікація на production, Масштабування, Роутінг, Управління станом програми, SSR. Відображення на стороні сервера, Безпека

##### Тема 10. Вступ до Node.js.

Що таке Node.js? Початок роботи. Перший додаток. Основи роботи з Node.js. Модулі. Робота з модулями. Об'єкт global та глобальні змінні. Передача параметрів додатку. NPM. Package.json. Встановлення модулів. Визначення команд. Nodemon. Асинхронність у Node.js. Робота з файлами. Події. Stream. Pipe. Створення сервера. Надсилання файлів. Отримання даних від клієнта. Шаблони

Тема 11. *Node.js. Express*

Початок роботи з Express. Конвеєр обробки запиту та middleware. Надсилання відповіді. Статичні файли. Маршрутизація. Переадресація. Передача даних програмі. Параметри рядка запиту. POST-запити та відправлення форм. Параметри маршруту. Router. JSON та AJAX. Уявлення та двигун уявлень Handlebars. Часткові уявлення у Handlebars. Layout в Handlebars. Хелпери в Handlebars. Двигун EJS. Двигун Pug. Часткові уявлення та layout в Pug. Створення API

Тема 12. *Node.js i MongoDB*

Початок роботи з MongoDB. Додавання даних до MongoDB. Отримання даних у MongoDB. Видалення документів у MongoDB. Оновлення документів у MongoDB. Express та MongoDB. Mongoose. Визначення схеми у Mongoose. CRUD у Mongoose. Express та Mongoose

Тема 13. *GraphQL API Server в Node.js*

Налаштування GraphQL HTTP Server Middleware, Використання GraphQL IDE, Запит GraphQL API від клієнта

**C. LIST OF COMPETENCIES AND STUDIES TARGETED RESULTS / ПЕРЕЛІК КОМПЕТЕНТНОСТЕЙ ТА ПРОГРАМНИХ РЕЗУЛЬТАТІВ НАВЧАННЯ**

Загальні компетентності (ЗК)	<p>ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК-2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК-3. Здатність проведення теоретичних та прикладних досліджень на відповідному рівні</p> <p>ЗК-8. Знати класифікацію програмного забезпечення та призначення його складових частин.</p> <p>ЗК-9. Здатність використовувати сучасні інтегровані прикладні програмні системи для обробки числової, текстової та графічної інформації.</p>
Спеціальні (фахові) компетентності (ФК)	<p>СК 2. Здатність розробляти архітектури, модулі та компоненти програмних систем.</p> <p>СК 3. Знання і розуміння специфікацій, стандартів, правил і рекомендацій в професійній галузі, уміння оцінювати ступінь обґрунтованості їх застосування, здатність дотримуватися їх при реалізації процесів життєвого циклу.</p> <p>СК 4. Здатність забезпечувати технічну підтримку і навчання користувачів програмного забезпечення.</p> <p>СК 5. Уміння готувати та презентувати документацію та методичні матеріали щодо програмного забезпечення.</p> <p>СК 6. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.</p> <p>СК 7. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності загальної функціональності і надійності програмного забезпечення.</p> <p>СК 8. Здатність до алгоритмічного та логічного мислення.</p> <p>СК 9. Здатність приймати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p>
Програмні результати навчання (ПРН)	<p>ПРН 2. Розуміти, аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.</p> <p>ПРН-3. Знати, розуміти і застосовувати ефективні підходи щодо проектування програмного забезпечення.</p> <p>ПРН 7. Знати, розуміти, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.</p> <p>ПРН-8. Знати, розуміти і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення.</p> <p>ПРН-11. Набувати нові наукові і професійні знання, вдосконалювати навички, прогнозувати розвиток програмних систем та інформаційних технологій.</p> <p>ПРН 12. Знати і уміти застосовувати інформаційні технології обробки, зберігання та передачі даних.</p> <p>ПРН-15. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.</p>

**D. SEMESTER PLAN / СЕМЕСТРОВИЙ ПЛАН**

Тиждень/ Дата	Тема, план/короткі тези	Форма діяльності (заняття), години, формат	Завдання для СРС (література, ресурси в інтернеті, презентація, відеокурси)
------------------	-------------------------	--	---



1-4 тиждень	<i>Тема 1. Основи JavaScript.</i> Структура коду, Суворий режим - "use strict", Змінні, Типи даних, Взаємодія: alert, prompt, confirm, Перетворення типів, Базові оператори, математика, Оператори порівняння, Умовне розгалуження: if, '?', Логічні оператори, Оператор нульового злиття (??), Цикли while та for, Конструкція "switch", Функції, Function Expression, Стрілкові функції	Лекція, (4 год), F2F	Опрацювання літератури: основна 1, 3, 5 додаткова 3
		Лабораторні заняття (4 год), F2F	Теми до вивчення: Перетворення типів, Базові оператори, математика, Оператори порівняння, Умовне розгалуження: if, '?', Логічні оператори, Лабораторна робота №1
5-6 тиждень	<i>Тема 2. Основи об'єктів.</i> Об'єкти, Копіювання об'єктів та посилання, Складання сміття, Методи об'єкта, "this", Конструктор, оператор "new", Опціональний ланцюжок '?.', Тип даних Symbol, Перетворення об'єктів у примітиви.	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 3, 5 додаткова 3
		Лабораторні заняття (2 год), F2F	Теми до вивчення: Методи об'єкта, "this", Конструктор, оператор "new", Опціональний ланцюжок '?.', Лабораторна робота №2
7-8 тиждень	<i>Тема 3. Типи даних.</i> Методи примітивів, Числа,Рядки, Масиви, Методи масивів, Об'єкти, що перебираються, Map та Set, WeakMap та WeakSet, Object.keys, values, entries, Деструктурируюче присвоєння, дата та час, Формат JSON, метод toJSON	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3, 5, 8 додаткова 1,2,3
		Лабораторні заняття (2 год), F2F	Теми до вивчення: Числа, Рядки, Масиви, Методи масивів, Об'єкти Лабораторна робота №3
9-10 тиждень	<i>Тема 4. Функції.</i> Рекурсія та стек, Залишкові параметри та оператор розширення, Замикання, Застаріле ключове слово "var", Глобальний об'єкт, Об'єкт функції, NFE, Синтаксис "New Function", Планування: setTimeout та setInterval, Декоратори та переадресація виклику, call/apply, Прив'язка контексту до функції, Повторюємо стрілчні функції	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3, 5, 7, 8 додаткова 1, 2, 3
		Лабораторні заняття (2 год), F2F	Теми до вивчення: Залишкові параметри та оператор розширення, Замикання, Застаріле ключове слово "var", Глобальний об'єкт, Об'єкт функції, Лабораторна робота №4
11-12 тиждень	<i>Тема 5. Класи.</i> Клас: базовий синтаксис, Наслідування класів, Статичні властивості та методи, Приватні та захищені методи та властивості, Розширення вбудованих класів, Перевірка класу: "instanceof", Домішки	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 5, 7, 8 додаткова 1,2,3,4
		Лабораторні заняття (2 год), F2F	Теми до вивчення: Наслідування класів, Статичні властивості та методи, Приватні та захищені методи та властивості, Розширення вбудованих класів Лабораторна робота №5
13-14 тиждень	<i>Тема 6. Проміси, async/await.</i> Колбеки, Промісли, Ланцюжок промісів, Промісли: обробка	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4



	помилки, Promise API, Промісіфікація, Мікрозавдання, Async/await	Лабораторні заняття (2 год), F2F	Теми до вивчення: Промисли, Ланцюжок промісів, Промисли: обробка помилок, Promise API, Async/await Лабораторна робота №6
15-18 тиждень	Тема 7. <i>Vue.js</i> , вступ. Встановлення, Вступ, Що таке <i>Vue.js</i> ?, Початок роботи, Декларативне малювання, Умови та цикли, Робота з введенням користувача, Розбиття програми на компоненти, Ставлення до елементів користувача <i>Web Components</i> ., Примірник <i>Vue</i> , Синтаксис шаблонів, Обчислювані властивості та стеження, Робота з класами та стилями, Умовне відмальовування, Відображення списків, Обробка подій, Робота з формами	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4
		Лабораторні заняття (4 год), F2F	Теми до вивчення: Початок роботи, Декларативне малювання, Умови та цикли, Робота з введенням користувача, Розбиття програми на компоненти, Лабораторна робота №7
19-20 тиждень	Тема 8. <i>Vue.js</i> , компоненти. Основи компонентів, Продвинуті компоненти, Реєстрація компонентів, Вхідні параметри, Події користувача, Слоти, Динамічні та асинхронні компоненти, Обробка крайніх випадків, Переходи та анімації, Анімація списків та появи/зникнення, Анімація переходів між станами	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4
		Лабораторні заняття (2 год), F2F	Теми до вивчення: Продвинуті компоненти, Реєстрація компонентів, Вхідні параметри, Події користувача, Слоти, Динамічні та асинхронні компоненти, Лабораторна робота №8
21-22 тиждень	Тема 9. <i>Vue.js</i> . <i>Перевикористання та композиція</i> Домішки, Директиви користувача, <i>Render</i> -функції та <i>JSX</i> , Плагіни, Фільтри, Інструментарій, Однофайлові компоненти, Тестування, Підтримка <i>TypeScript</i> , Публікація на <i>production</i> , Масштабування, Роутінг, Управління станом програми, <i>SSR</i> . Відображення на стороні сервера, Безпека	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4
		Лабораторні заняття (2 год), F2F	Теми до вивчення: <i>Render</i> -функції та <i>JSX</i> , Плагіни, Фільтри, Інструментарій, Однофайлові компоненти, Тестування, Підтримка <i>TypeScript</i> , Лабораторна робота №9
23-24 тиждень	Тема 10. <i>Вступ до Node.js</i> . Що таке <i>Node.js</i> ? Початок роботи. Перший додаток. Основи роботи з <i>Node.js</i> . Модулі. Робота з модулями. Об'єкт <i>global</i> та глобальні змінні. Передача параметрів додатку. <i>NPM</i> . <i>Package.json</i> . Встановлення модулів. Визначення команд. <i>Nodemon</i> . Асинхронність у <i>Node.js</i> . Робота з файлами. Події.	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4
		Лабораторні заняття (2 год), F2F	Теми до вивчення: Основи роботи з <i>Node.js</i> . Модулі. Робота з модулями. Об'єкт <i>global</i> та глобальні змінні. Передача параметрів додатку. <i>NPM</i> . <i>Package.json</i> . Встановлення модулів. Визначення команд.



	Stream. Pipe. Створення сервера. Надсилання файлів. Отримання даних від клієнта. Шаблони		Nodemon. Лабораторна робота №10
25-28 тиждень	Тема 11. <i>Node.js. Express</i> Початок роботи з Express. Конвеєр обробки запиту та middleware. Надсилання відповіді. Статичні файли. Маршрутизація. Переадресація. Передача даних програмі. Параметри рядка запиту. POST-запити та відправлення форм. Параметри маршруту. Router. JSON та AJAX. Уявлення та двигун уявлень Handlebars. Часткові уявлення у Handlebars. Layout в Handlebars. Хелпери в Handlebars. Двигун EJS. Двигун Pug. Часткові уявлення та layout в Pug. Створення API	Лекція, (4 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4
		Лабораторні заняття (4 год), F2F	Теми до вивчення: Статичні файли. Маршрутизація. Переадресація. Передача даних програмі. Параметри рядка запиту. POST-запити та відправлення форм. Параметри маршруту. Router. Лабораторна робота №11
29-30 тиждень	Тема 12. <i>Node.js і MongoDB</i> Початок роботи з MongoDB. Додавання даних до MongoDB. Отримання даних у MongoDB. Видалення документів у MongoDB. Оновлення документів у MongoDB. Express та MongoDB. Mongoose. Визначення схеми у Mongoose. CRUD у Mongoose. Express та Mongoose	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4
		Лабораторні заняття (2 год), F2F	Теми до вивчення: MongoDB. Додавання даних до MongoDB. Отримання даних у MongoDB. Видалення документів у MongoDB. Оновлення документів у MongoDB. Express та MongoDB. Mongoose Лабораторна робота №12
31-32 тиждень	Тема 13. <i>GraphQL API Server в Node.js</i> Налаштування GraphQL HTTP Server Middleware, Використання GraphQL IDE, Запит GraphQL API від клієнта	Лекція, (2 год), F2F	Опрацювання літератури: основна 1, 2, 3,4, 8 додаткова 1,2,3,4
		Лабораторні заняття (2 год), F2F	Теми до вивчення: Налаштування GraphQL Використання GraphQL IDE Лабораторна робота №13
33-34 тиждень	Заняття для захисту індивідуальних робіт.	Лабораторне заняття (2 год), F2F	Виправлення недоліків індивідуальних робіт, виконання програмного коду, розрахунок контрольного прикладу

Вивчення дисципліни передбачає виконання двох аудиторних тестових завдань за допомогою програми Zelis. Перша контрольна модульна робота виконується за темами 1-4, друга – за темами 5-8 у тестовій формі. Під час виконання студенти мають продемонструвати уміння та навички залучати набуті теоретичні та практичні знання з програмування на мові Python.

Замість науково-дослідницької роботи студенти можуть отримати 10 балів за вивчення онлайн курсу на платформі COURSERA

Детальний план проведення лабораторних занять, завдання для індивідуальних робіт містяться в системі MOODLE у відповідних розділах.

#### E. BASIC LITERATURE (OBLIGATORY TEXTBOOKS) / ОСНОВНА ЛІТЕРАТУРА (ОБОВ'ЯЗКОВІ ПІДРУЧНИКИ)

1. Vue.JS [Електронний ресурс]. – Режим доступу : <https://ru.vuejs.org/v2/guide/index.html>.
2. ArcGIS Pro Python [Електронний ресурс]. – Режим доступу : <https://pro.arcgis.com/ru/pro-app/arcpy/main/arcgis-pro-arcpy-reference.htm>.
3. Python. Обучение программированию [Электронный ресурс]. – Режим доступа : <https://younglinux.info/python>
4. Python ООП: <https://python-scripts.com/object-oriented-programming-in-python#pros-cons-oop>
5. Флэнаган Д. JavaScript. Подробное руководство, 6-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2012. –



1080 с..

6. Дэвид Херрон. Node.js Web Development – 4 издание -
7. Дэвид М. Бизли Python. Подробный справочник / Дэвид М. Бизли. – СПб. : Символ-Плюс, 2010. – 864 с.
8. Саммерфилд М. Программирование на Python 3. Подробное руководство / М. Саммерфилд. – СПб. : Символ-Плюс, 2009. – 608 с.
9. Саммерфилд М. Python на практике / М. Саммерфилд – М. : ДМК Пресс, 2014. – 338 с.
10. Сузи Р. А. Язык программирования Python : учеб. пособие / Р. А. Сузи. – М. : ИНТУИТ, БИНОМ. Лаборатория знаний, 2006. – 328 с.
11. Доусон М. Програмуємо на Python / М. Доусон. – СПб. : Питер, 2012. – 432 с.
12. Хахаев И. А. Практикум по алгоритмизации и программированию на Python: учебник / И. А. Хахаев. – М. : Альт Линукс, 2010. – 126 с.

#### F. COMPLEMENTARY LITERATURE / ДОДАТКОВА ЛІТЕРАТУРА

1. А. Мюллер, С. Гвидо - Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными – М. 2017. — 393 с.
2. Буйначев С.К., Боклаг Н.Ю. - Основы программирования на языке Python – Екатеринбург. - 2014. — 90 с.
3. Бэрри П. - Изучаем программирование на Python (Мировой компьютерный бестселлер). - М.- 2017. — 618 с.
4. Зед Шоу - Легкий способ выучить Python (Мировой компьютерный бестселлер) - 2017. — 353 с.
5. Федоров Д. - Основы программирования на примере языка Python - М.- 2018. – 167 с.  
Чан Уэсли Дж. - Python. создание приложений (Библиотека профессионала) –М. – 2015.- 794 с.

#### G. THE MOST IMPORTANT PUBLICATIONS OF THE AUTHOR(S) CONCERNING PROPOSED CLASSES / ОСНОВНІ ПУБЛІКАЦІЇ АВТОРА, ЩО ПОВ'ЯЗАНІ З ТЕМАТИКОЮ ЗАПЛАНОВАНИХ ЗАНЯТЬ

1.

#### H. PREREQUISITE AND POSTREQUISITE / ПРЕРЕКВІЗИТИ ТА ПОСТРЕКВІЗИТИ

#### I. SCOPE AND TYPE / КІЛЬКІСТЬ ВІДВЕДЕНИХ ГОДИН ТА ФОРМА ПРОВЕДЕННЯ ЗАНЯТЬ

	Денна	Заочна
Лекції	34	8
Практичні (лабораторні)	34	8
Самостійна робота студента (СРС)	76	128
Індивідуально-консультативна робота (ІКР)	36	36
Курсова робота	–	–

#### J. CURRENT AND FINAL EVALUATION / ПОТОЧНЕ ТА ПІДСУМКОВЕ ОЦІНЮВАННЯ

	Денна	Заочна
Поточний контроль, в т.ч.:	50	50
оцінювання під час аудиторних занять	10	10
виконання контрольних (тестових) робіт	20	20
виконання і захист завдань самостійної роботи	60	60
науково-дослідницька робота	10	10
Підсумковий контроль (екзамен)	-	-
Разом	100	100

#### J. CURRENT AND FINAL EVALUATION / ПОТОЧНЕ ТА ПІДСУМКОВЕ ОЦІНЮВАННЯ

	Денна	Заочна
Шкала балів	Оцінка за 4-бальною шкалою	Шкала ECTS
90 – 100	Відмінно	A
80 – 89	Добре	B
70 – 79		C
66 – 69		D
60 – 65	Задовільно	E
21 – 59	незадовільно з можливістю повторного складання екзамену (заліку)	FX
0 – 20	незадовільно з можливістю вивчення дисципліни за індивідуальним графіком у формі додаткової індивідуально-консультативної роботи.	F

#### K. CODE OF CONDUCT OF THE COURSE / КОДЕКС ПОВЕДІНКИ ПІД ЧАС ВИВЧЕННЯ КУРСУ

Для успішного проходження курсу та складання контрольних заходів необхідним є виконання наступних

обов'язків:

- не запізнюватися на заняття;
- не пропускати заняття (як лекційні, так і практичні), в разі хвороби мати довідку або її ксерокопію;
- самостійно опрацювати весь лекційний матеріал та ресурси для самостійної роботи;
- конструктивно підтримувати зворотній зв'язок з викладачем на всіх етапах проходження курсу (особливо під час виконання індивідуальних проектів/курсів проекту);
- своєчасно і самостійно виконувати всі передбачені програмою лабораторні завдання;
- брати очну участь у контрольних заходах;
- будь-яке відтворення результатів чужої праці (включаючи практичну роботу над командним проектом), в тому числі використання завантажених з Інтернету матеріалів, як власних результатів, кваліфікується, як порушення норм і правил академічної доброчесності, та передбачає притягнення до відповідальності у порядку, визначеному чинним законодавством.

#### L. METHODS OF CONDUCTING / МЕТОДИ НАВЧАННЯ

Для формування компетентностей застосовуються такі методи навчання:

*вербальні/словесні* (лекція, пояснення, дискусія);

*наочні* (спостереження, ілюстрація, демонстрація, відео-уроки);

*практичні* (різні види лабораторних завдань, вирішення задач, проведення експерименту, практики);

*пояснювально-ілюстративний*, який передбачає пред'явлення готової інформації викладачем та її засвоєння студентами;

*метод проблемного викладу*;

*дослідницький*.

#### M. TOOLS, EQUIPMENT AND SOFTWARE / ІНСТРУМЕНТИ, ОБЛАДНАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

мультимедійний клас з ПК, цифровий проектор, інтегрована серва IDLE Python

[Zoom](#) – хмарна платформа для відео і аудіо конференцій та вебінарів

Тестування знань та отримання оцінки з екзамену:

ZELIS - система призначена для тестування знань студентів в двох режимах: автоматизований контроль знань та тестування по бланкам.

#### N. STUDENT RESOURCES, MOOC PLATFORMS / ЦИФРОВІ РЕСУРСИ ДЛЯ СТУДЕНТІВ ТА ВІДКРИТІ ДИСТАНЦІЙНІ ОНЛАЙН КУРСИ

Студентам пропонується доступ до навчальних матеріалів дисципліни - <http://moodle.kneu.dp.ua>.

[Coursera](#) – безкоштовні онлайн-курси з різних дисциплін, у разі успішного закінчення яких користувач отримує сертифікат про проходження курсу.

[EdX](#) – онлайн-курси від закладів вищої освіти.

[Prometheus](#) – український громадський проект масових відкритих онлайн-курсів.

#### O. FEEDBACK/ ЗВОРОТНІЙ ЗВ'ЯЗОК

Електронні листи є найкращим способом зв'язатися з керівником курсу, і, будь ласка, додайте шифр групи в темі листа. Якщо ви надішлете мені електронне повідомлення, надайте мені, принаймні, 48 годин (протягом робочого тижня), щоб відповісти. Якщо ви не отримуєте відповідь, відправте листа повторно.

#### P. ACADEMIC HONESTY/ АКАДЕМІЧНА ДОБРОЧЕСНІСТЬ

Державний університет економіки і технологій очікує від студентів розуміння та підтримання високих стандартів академічної чесності. Приклади академічної не доброчесності включають такі: плагіат, зловживання інформацією із застарілих джерел мережі. Очікується, що вся робота, виконана відповідно до вимог курсу, є власною роботою студента. Під час підготовки роботи, яка відповідає вимогам курсу, студенти повинні відрізняти власні ідеї від інформації, отриманої з інших джерел. Без попереднього письмового схвалення викладачем, студенти можуть не подавати один і той же звіт двічі.

#### APPROVED / ЗАТВЕРДЖЕНО

Рішенням кафедри інформатики і ППЗ Державного університету економіки і технологій - протокол №\_\_ від \_\_.\_\_.2022 року

Укладач:

Дмитро МЕДВЕДСВ

ЗАТВЕРДЖЕНО:

Кафедрою інформатики і прикладного програмного  
забезпечення

Протокол № 1 від 25 серпня 2022 року

В.о. завідувача кафедри



Олександр ЗЕЛЕНСЬКИЙ

Науково-методичною радою Державного університету  
економіки і технологій

Протокол № 1 від 20 вересня 2022 року

Голова науково-методичної ради



Валентин ОРЛОВ