



## Сучасні технології розробки WEB-застосувань мовою програмування PHP Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення, 123 Комп'ютерна інженерія, 126 Інформаційні системи та технології</i>
Освітня програма	<i>Інженерія програмного забезпечення інформаційних систем, Інженерія програмного забезпечення комп'ютерних систем, Інтегровані інформаційні системи, Інформаційні управляючі системи та технології, Інформаційне забезпечення робототехнічних систем, Комп'ютерні системи та мережі</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>3 курс, осінній семестр</i>
Обсяг дисципліни	<i>4 кредити / 120 год (36 годин лекцій, 18 годин комп'ютерних практикумів (лабораторних робіт), 66 годин самостійна робота)</i>
Семестровий контроль/ контрольні заходи	<i>Залік/ Модульні контрольні роботи</i>
Розклад занять	<i><a href="https://my.kpi.ua">https://my.kpi.ua</a> <a href="https://schedule.kpi.ua">https://schedule.kpi.ua</a></i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: ст.викладач Олесь Володимирович Ковтунець, <a href="mailto:kovtunets.oles@ill.kpi.ua">kovtunets.oles@ill.kpi.ua</a>, Комп'ютерні практикуми (лабораторні роботи): ст.викладач Олесь Володимирович Ковтунець, <a href="mailto:kovtunets.oles@ill.kpi.ua">kovtunets.oles@ill.kpi.ua</a>, к.т.н., доцент Губський Андрій Миколайович, <a href="mailto:andrew.gubskiy@gmail.com">andrew.gubskiy@gmail.com</a> .</i>
Розміщення курсу	<i><a href="https://classroom.google.com/c/ODAYNzk1NTU3OTM5?cjc=cbedrek4">https://classroom.google.com/c/ODAYNzk1NTU3OTM5?cjc=cbedrek4</a></i>

### Програма навчальної дисципліни

#### 1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Вивчення дисципліни спрямовано на оволодіння сучасними технологіями розробки веб-застосувань мовою програмування PHP на основі теоретичних знань про застосування шаблонів проектування та принципів їх використання при розробці веб-застосувань та спеціалізованого програмного забезпечення для роботи із базовими технологіями структурування, візуалізації та забезпечення взаємодії веб-застосувань у відповідності із діючими веб-стандартами, а також на вивчення особливостей, процесів організації роботи та підходів до веб-розробки, що дає змогу виявляти

проблеми швидкодії та оптимізувати функціонування веб-застосувань, розроблених з використанням мови програмування PHP, а також власноруч створювати веб-застосування мовою програмування PHP. Дисципліна розвиває та удосконалює професійні навички студентів, набуті в попередні роки навчання.

**Предмет** навчальної дисципліни – методи і засоби структурування, візуалізації та забезпечення взаємодії вебзастосунків, технології, засоби і методи використання веброзробки мовою програмування PHP, способи проектування, розробки, впровадження та супроводження вебзастосунків мовою програмування PHP із урахуванням актуальності та перспективності супутніх технологій веброзробки.

**Метою** дисципліни є отримання студентами фундаментальних знань про принципи побудови, функціонування та обслуговування вебзастосунків із використанням мови програмування PHP, дослідження особливостей традиційних і перспективних методик веброзробки мовою програмування PHP, формування розуміння суті та взаємодії процесів у робочому довкіллі PHP-програміста, набуття ключових фахових компетентностей, теоретичних знань і практичних навичок з технологій розробки вебзастосунків мовою програмування PHP, та проектування, розробка, впровадження та супроводження вебзастосунків мовою програмування PHP.

За результатами вивчення дисципліни студент повинен набути **досвід** з розробки вебзастосунків мовою програмування PHP у відповідності із діючими вебстандартами, адекватного моделювання предметних областей, створення сучасних вебзастосунків мовою програмування PHP, використання професійно профільованих знань та умінь в практичному використанні інтернет-технологій та технологій розробки вебзастосунків мовою програмування PHP, на основі взаємодії користувача і вебзастосунку та вебзастосунку і серверних компонентів вміти аналізувати роботу та виявляти помилки, вміти оптимізувати існуючі PHP-застосунки та проектувати нові вебзастосунки з урахуванням потреб потенційної аудиторії, аналізувати, аргументувати, приймати рішення при проектуванні, розробці, впровадженні та супроводженні вебзастосунків в умовах чітко встановлених або нечітких (загальних) заданих технологічних та законодавчих технічних завдань.

На комп'ютерних практикумах студенти опанують етапи проектування PHP-застосунків і використання HTML, CSS, Javascript та PHP, включаючи створення PHP-застосунків для різних типів програмного забезпечення та пристроїв.

В курсі передбачений контроль якості отриманих знань у вигляді модульної контрольної роботи.

**Програмні результати навчання** студента. Згідно з вимогами освітньо-професійної програми студент після засвоєння навчальної дисципліни повинен **знати** такі категорії:

- Інтерфейс користувача.
- Шаблон проектування PHP-застосунку.
- Стандарти та специфікації мови програмування PHP.
- Клієнтські мови програмування в контексті взаємодії із серверними застосунками.
- Використання фреймворків мови програмування PHP.
- Особливості розробки вебзастосунків мовою програмування PHP.
- Створення інтерактивних вебзастосунків мовою програмування PHP.
- Обмін даними в синхронному та асинхронному режимах між веб-застосунками.
- Використання баз даних та сховищ даних.
- Інформаційна гігієна та захист персональних даних.
- Розробка безпечних вебзастосунків мовою програмування PHP.

- Способи злому та захисту PHP-застосунків.
- Оптимізація розроблюваних PHP-застосунків з урахуванням доступних технологій та апаратних засобів.

Студент повинен **вміти**:

- використовувати мову програмування PHP для написання ефективних програм,
- застосовувати принципи об'єктно-орієнтованого програмування у розробці програмних продуктів,
- працювати з PHP-бібліотеками для виконання стандартних задач,
- розробляти графічний інтерфейс користувача,
- використовувати механізми та інструменти багатопотоковості у розробці програмного забезпечення.

Вивчення дисципліни спрямоване на оволодіння студентом таких компетентностей: Здатність проєктувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

**Програмні результати** вивчення дисципліни забезпечують такі програмні результати освітньої програми:

Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проєктування і використання інформаційних систем та технологій.

Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.

Обґрунтовувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.

## **2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)**

Вивчення дисципліни спирається на базові знання з основ програмування, алгоритмів та структур даних, а також на знання, уміння і навички, необхідні для раціонального використання засобів інформаційних технологій при розв'язуванні задач опрацювання інформації, розуміння ролі інформаційних технологій, основ інформаційної культури та гігієни. Знання та навички, набуті студентом при вивченні дисципліни, використовуються в розробці веб-документів та веб-застосовань на їхній основі, в розробці дипломних проєктів здобувачів першого (бакалаврського) рівня вищої освіти за освітніми програмами спеціальностей: 121 Інженерія програмного забезпечення 123 Комп'ютерна інженерія, 126 Інформаційні системи та технології.

## **3. Зміст навчальної дисципліни**

### **Розділ 1. Мережі, обмін даними, візуалізація, проблематика**

Тема 1.1. Організація мережі, поняття, протоколи та тенденції.

Тема 1.2. Дані: формати обміну та способи візуалізації.

Тема 1.3. Інтерфейс користувача: фреймворки для швидкого проєктування.

Тема 1.4. Швидкість розвитку вебу та пов'язані з цим проблеми.

### **Розділ 2. Організація вебпрограмування**

Тема 2.1. Системи контролю версій та спільна робота команди.

- Тема 2.2. Особливості веброзробки мовою програмування PHP.
- Тема 2.3. Клієнтські мови програмування в контексті взаємодії із серверними застосуваннями.
- Тема 2.4. Засоби автоматизації.
- Тема 2.5. Підвищення ефективності веброзробки мовою програмування PHP.
- Тема 2.6. Прикладний програмний інтерфейс.
- Розділ 3. Серверна мова програмування PHP.**
- Тема 3.1. Стандарти та специфікації PHP.
- Тема 3.2. Особливості PHP.
- Тема 3.3. Розробка ігрових додатків.
- Тема 3.4. Розмаїття веброзробки мовою програмування PHP.
- Тема 3.5. Бази даних в інтернеті.
- Тема 3.6. Особливості клієнт-серверної взаємодії для мови програмування PHP.
- Розділ 4. Ріст та поширення.**
- Тема 4.1. Оптимізація використання серверних ресурсів.
- Тема 4.2. Питання безпеки PHP-застосунків.

## 4. Навчальні матеріали та ресурси

### Основна література

1. R. Crowther, J. Lennon, A. Blue, G. Wanish (2014). HTML5 in Action. Manning Publications Co. ISBN 978-1617290497.
2. Вчимо веб-розробку. – [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/uk/docs/Learn>
3. R. Nixon (2014). Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites. O'Reilly. ISBN 978-1491949467.
4. Р. Мельник (2018). Програмування веб-застосунків (фронт-енд та бек-енд). Львівська політехніка. ISBN 978-966-941-195-2.
5. Matt Zandstra (2021). PHP 8 Objects, Patterns, and Practice: Mastering OO Enhancements, Design Patterns, and Essential Development Tools. Apress. ISBN 978-1484267905.
6. Pro Git. – [Електронний ресурс]. – Режим доступу: <https://git-scm.com/book/ru/v2>
7. The State of JavaScript. – [Електронний ресурс]. – Режим доступу: <https://2024.stateofjs.com/>

### Додаткова література

8. Productivity for developers, performance for users. – [Електронний ресурс]. – Режим доступу: <https://www.gwtproject.org/>
9. S. Moreto. Bootstrap by Example. – Packt Publishing, 2016.
10. Т. Бернерс-Лі, М. Фічетті. Заснування п@вутини. З чого починалася і до чого прийде всесвітня мережа. – Києво-Могилянська академія, 2007.
11. The World's Largest Web Developer Site. – [Електронний ресурс]. – Режим доступу: <https://www.w3schools.com/>

12. K, Daimi. Computer and Network Security Essentials. – Springer International Publishing, 2018.

13. Закон України "Про захист персональних даних". – [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2297-17#Text>

Для викладання дисципліни необхідні наступні ресурси:

- в лекційній аудиторії має бути комп'ютер з доступом до мережі інтернет, а також проектор;
- в аудиторії, де проводяться комп'ютерні практикуми, мають бути робочі станції з доступом до мережі інтернет і браузерями, в кількості студентів у групі, для проходження модульної контрольної роботи;
- на робочих станціях має бути встановлене наступне вільне програмне забезпечення:
  - веб-сервер Apache,
  - СУБД MySQL або MariaDB,
  - інтерпретатор мови програмування PHP.

## 5. Методика опанування навчальної дисципліни (освітнього компонента)

Матеріали для вивчення дисципліни розміщені викладачем на платформі Google Classroom <https://classroom.google.com/c/ODAyNzk1NTU3OTM5?cjc=cbedrek4>, до якого є доступ студентів. Контент платформи доступний із будь-якого місця в мережі Інтернет. Навчальний курс вміщає теоретичні матеріали, матеріали для виконання завдань комп'ютерних практикумів, та матеріали для самостійної роботи.

Лекції з дисципліни проводяться викладачем із використанням сучасних мультимедійних презентаційних технологій. На лекціях проводяться експрес-опитування, які дають можливість лектору отримати інформацію про якість засвоєння матеріалу та, за необхідності, розглянути більш детально складний матеріал.

Завдання комп'ютерних практикумів виконуються з використанням обраного редактора коду та при потребі веб-сервера, системи контролю версій Git та броузера, з перевіркою функціональності на різних типах пристроїв (комп'ютер і смартфон).

Під час проведення комп'ютерних практикумів використовуються методичні вказівки до виконання завдань комп'ютерних практикумів з дисципліни. Студент на початку семестру отримує календарний план початку та завершення захисту виконаних робіт.

Модульні контрольні роботи містять завдання з перевірки як теоретичних знань, так і практичних навичок, і проводяться на онлайн-заняттях за допомогою Zoom.

### 5.1. Тематика лекцій

№ з/п	Назва теми лекції та перелік основних питань (дидактичні матеріали: презентації Power Point або PDF, відеолекції)
1	<b>Розділ 1. Мережі, обмін даними, візуалізація, проблематика.</b> <b>Тема 1.1. Організація мережі, поняття, протоколи та тенденції.</b> Організація мережі. Основні поняття мережі. Масштабування мереж. Мережеві протоколи. Глобальні мережі. Сучасні тенденції глобального розвитку і локальних особливостей Інтернету. Література: 1, 3, 10.
2	<b>Тема 1.2. Дані: формати обміну та способи візуалізації.</b> Розвиток форматів обміну даними та способів візуалізації даних. Структурування даних, формати обміну даними. Автоматизація роботи зі структурою вебзастосунку. Література: 2, 3, 4.
3	<b>Тема 1.3. Інтерфейс користувача: фреймворки для швидкого проєктування.</b> Дизайн інтерфейсу користувача. Основи дизайну. Швидка розробка інтерфейсу користувача за допомогою фреймворків. Робота з графікою. Література: 1, 3, 9.
4	<b>Тема 1.4. Швидкість розвитку вебу та пов'язані з цим проблеми.</b> Когнітивні проблеми у вебi. Швидкість розвитку вебу. Когнітивні проблеми користувачів сучасного ІТ-сектору. Взаємодія з кінцевим користувачем вебзастосунку. Література: 1, 4, 10.

5	<p><b>Розділ 2. Організація вебпрограмування.</b>  <b>Тема 2.1. Системи контролю версій та спільна робота команди.</b>  Системи контролю версій. Типи систем контролю версій. Організація гілок.  Спільна робота команди.  Література: 6.</p>
6	<p><b>Тема 2.2. Особливості веброзробки мовою програмування PHP.</b>  Парадигми та технології для розробки вебзастосунків. Походження та класифікація методологій. Врахування особливостей веброзробки мовою PHP для вибору методології. Планування веброзробки в рамках обраних методологій та технологій.  Література: 4, 8.</p>
7	<p><b>Тема 2.3. Клієнтські мови програмування в контексті взаємодії із серверними застосунками.</b>  Клієнтські мови програмування. Серверні мови програмування. Стандарти та специфікації клієнт-серверної взаємодії. Способи створення сценаріїв взаємодії.  Література: 2, 3, 4, 5.</p>
8	<p><b>Тема 2.4. Засоби автоматизації.</b>  Автоматизація рутини для уникнення механічних помилок. Використання систем контролю версій. Використання засобів автоматичного тестування.  Література: 3, 4, 5.</p>
9	<p><b>Тема 2.5. Підвищення ефективності веброзробки мовою програмування PHP.</b>  Підвищення ефективності веброзробки. Повторне використання коду.  Використання бібліотек та фреймворків.  Література: 3, 4, 5, 8.</p>
10	<p><b>Тема 2.6. Прикладний програмний інтерфейс.</b>  Прикладний програмний інтерфейс. Методи взаємодії вебкомпонентів. Способи обміну даними.  Література: 3, 4, 5, 8.</p>
11	<p><b>Розділ 3. Серверна мова програмування PHP.</b>  <b>Тема 3.1. Стандарти та специфікації PHP.</b>  Використання PHP та розширень для веброзробки. Стандарти, специфікації, особливості. Взаємодія з довідками, з користувачем, з сервером, між вебзастосунками.  Література: 2, 3, 4, 5.</p>
12	<p><b>Тема 3.2. Особливості PHP.</b>  Використання особливостей PHP. Режими функціонування PHP-скриптів.  Використання пам'яті у PHP.  Література: 2, 3, 4, 5.</p>
13	<p><b>Тема 3.3. Розробка ігрових додатків.</b>  Використання можливостей візуалізаційних бібліотек та движків для ігрових застосунків на PHP.  Література: 2, 3, 4, 5.</p>
14	<p><b>Тема 3.4. Розмаїття веброзробки мовою програмування PHP.</b>  Варіативність веброзробки мовою програмування PHP. Особливості розробки вебзастосунків та односторінкових застосунків у прикладному програмному забезпеченні.  Література: 2, 3, 4, 5.</p>

15	<p><b>Тема 3.5. Бази даних в інтернеті.</b>  Використання баз даних. Структуроване зберігання даних. Локальні бази даних, клієнтські сховища даних. Особливості роботи баз даних на великих масивах даних.  Література: 2, 3, 4, 5.</p>
16	<p><b>Тема 3.6. Особливості клієнт-серверної взаємодії для мови програмування PHP.</b>  Взаємодія клієнта і сервера. Клієнтське та серверне використання веборієнтованих мов програмування.  Література: 2, 3, 4, 5.</p>
17	<p><b>Розділ 4. Ріст та поширення.</b>  <b>Тема 4.1. Оптимізація використання серверних ресурсів.</b>  Оптимізація вебзастосунків та довкілля виконання з урахуванням доступних технологій та апаратних засобів. Керування серверними ресурсами.  Література: 3, 4, 5.</p>
18	<p><b>Тема 4.2. Питання безпеки PHP-застосунків.</b>  Безпека в інтернеті. Принципи інформаційної гігієни. Персональні дані в Україні і світі. Пошук вразливої інформації в інтернеті. Тестування вебзастосунків для виявлення перевантажень, збоїв та вразливостей. Методики програмування для мінімізації можливостей злому. Розповсюдження вебзастосунків.  Література: 3, 4, 5, 12, 13.</p>

## 5.2. Тематика комп'ютерних практикумів/лабораторних робіт

№ з/п	Назва теми комп'ютерного практикуму та перелік основних питань (дидактичні матеріали: презентації Power Point або PDF, відеолекції)
1	<b>Комп'ютерний практикум 1. Розробка PHP-структур та вивід результатів у вебсторінку.</b> Мета: Ознайомлення з мовою програмування PHP. Література: 1, 2, 3, 4, 10.
2	<b>Комп'ютерний практикум 2. Взаємодія вебзастосунку із користувачем та локальним сховищем.</b> Мета: Ознайомлення із можливістю забезпечення інтерактивності вебзастосунку та зберігання даних у локальному сховищі броузера. Література: 1, 3, 4, 9, 10.
3	<b>Комп'ютерний практикум 3. Взаємодія вебзастосунку із базою даних.</b> Мета: Ознайомлення із можливістю зберігання структурованої інформації вебзастосунку у базі даних. Література: 4, 6, 8.
4	<b>Комп'ютерний практикум 4. Взаємодія фронтенду та бекенду.</b> Мета: Організація бекенду і фронтенду за допомогою Javascript та PHP. Література: 2, 3, 4, 5.
5	<b>Комп'ютерний практикум 5. Експерименти із підвищенням навантаженням.</b> Мета: Ознайомлення із роботою серверів та скриптів в режимі підвищеного навантаження. Література: 3, 4, 5, 8.

## 6. Самостійна робота студента

До самостійної роботи студента відноситься виконання завдань комп'ютерних практикумів, робота з документацією, опрацювання лекційного та додаткового теоретичного матеріалу за наданими презентаціями лекцій, навчальним посібником та додатковою літературою. Матеріали для самостійного вивчення дисципліни розміщені викладачем в електронному вигляді на платформі Google Classroom. Контент платформи доступний із будь-якого місця в мережі Інтернет. На самостійну роботу студент має витратити 66 годин.

### Самостійна робота

№ з/п	Назва розділу, теми (окремого питання), що виноситься на самостійне опрацювання
1	Пошук, оброблення та узагальнення інформації з різних джерел, критичне мислення, власна оцінка та аргументація для обраних ресурсів самоосвіти, перевірка роботоспроможності демонстраційних прикладів, підбір онлайн-ресурсів для тестування власного коду.
2	Написання коду в текстовому редакторі, вибір зручного редактора для роботи, тестування на локальному сервері, хостінг, розміщення в інтернеті.

3	Візуальний дизайн, структурування, організація та маркування, пошукові та керуючі елементи, проектування взаємодії, зручність та доступність, організація взаємодії користувача і системи, підбір зручних та незручних вебзастосунків, виявлення характеристик, що сприяють/заважають зручній роботі вебзастосунку, застосування принципів розробки зручного інтерфейсу, підготовка зображень у графічному редакторі.
4	Психологія та ІТ, когнітивне упередження, когнітивна напруга, когнітивне перевантаження, взаємодія розробника із користувачем.
5	Типи систем контролю версій, організація гілок, спільна робота команди.
6	Походження та класифікація методологій, врахування особливостей веброзробки для вибору методології, планування веброзробки в рамках обраних методологій та технологій.
7	Стандарти та специфікації веборієнтованих мов програмування, способи створення сценаріїв.
8	Використання систем контролю версій, використання засобів автоматичного тестування.
9	Повторне використання коду, використання бібліотек та фреймворків, власні бібліотеки коду, організація зберігання бібліотек коду.
10	Прикладний програмний інтерфейс, методи взаємодії вебкомпонентів, способи обміну даними.
11	Стандарти, специфікації, особливості, взаємодія з довіллям, з користувачем, з сервером, між вебзастосунками.
12	Особливості PHP, використання пам'яті у PHP.
13	Використання можливостей CSS та JS для створення анімаційних ефектів, canvas, розробка ігрових додатків.
14	Особливості розробки вебзастосунків та односторінкових застосунків у прикладному програмному забезпеченні.
15	Структуроване зберігання даних, локальні бази даних, клієнтські сховища даних, особливості роботи баз даних на великих масивах даних.
16	Клієнтське та серверне використання мови програмування JavaScript, встановлення та налаштування Node.js, підбір модулів для комфортної роботи.
17	Оптимізація вебзастосунків із урахуванням доступних технологій та апаратних засобів, серверні ресурси, клієнтські ресурси, оптимізація роботи з доступними проєктними потужностями.
18	Виявлення вразливостей у власному та у сусідському вебзастосунках, пошук рішень для захисту вебзастосунку від несанкціонованих дій. Порівняння способів та прийомів захисту персональних даних в Україні, Європі та США. Аналіз власної поведінки в інтернеті для виявлення небажаного розміщення персональних даних в інтернеті.

## Політика та контроль

### 7. Політика навчальної дисципліни (освітнього компонента)

Студент повинен вивчати дисципліну протягом семестру, дотримуючись рекомендованого календарного плану виконання комп'ютерних практикумів, вивчення тем лекційного матеріалу та виконання модульних контрольних робіт. Усі завдання студент має

виконувати самостійно і вчасно. Комп'ютерний практикум вважається виконаним, якщо студент захистив завдання практикума у викладача та розмістив звіт з виконання у відповідному розділі дисципліни на платформі Google Classroom.

При захистах комп'ютерних практикумів викладач дотримується такої політики справедливості:

1) студент, що захищає комп'ютерний практикум у відповідності з рекомендованим календарним планом, має вищий пріоритет у черзі на захист;

2) чим більше студент відстає від календарного плану захисту виконаних завдань, тим відповідно нижчим є його пріоритет у черзі на захист;

3) після захисту одного комп'ютерного практикуму на парі студент може захистити ще один лише після того, як усі присутні на занятті студенти захистили свій один комп'ютерний практикум.

Студент не має права претендувати на захист комп'ютерного практикуму поза межами пар, передбачених розкладом в семестрі. Можливість захистити комп'ютерний практикум надається в межах першої та другої перездач.

Дві модульні контрольні роботи проводяться в семестрі з використанням Zoom. Дату та час проведення робіт оголошує лектор на початку семестру, відмічає її у журналі та нагадує про проведення роботи на лекційних заняттях. Повторне виконання модульної контрольної роботи поза призначеним часом її виконання не передбачено.

Якщо студент не задоволений сумарною оцінкою за модульні контрольні роботи, він може переписати їх на семестровій контрольній роботі, що містить завдання з усього курсу.

Заохоченням до своєчасного засвоєння теоретичного матеріалу є бали за правильні відповіді на запитання під час експрес-опитування на лекціях.

Оцінювання студентів здійснюється згідно рейтингової оцінки рівня підготовки студентів з дисципліни. Поточний стан успішності студенти можуть бачити в електронному журналі.

Рейтингова система оцінювання з кредитного модуля описана у наступному розділі робочої програми.

## **8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)**

Поточний контроль: завдання комп'ютерних практикумів та МКР.

Календарний контроль: двічі на семестр як моніторинг поточного стану виконання вимог робочої програми.

Семестровий контроль: залік.

### **8.1 Поточний контроль**

Поточний контроль успішності засвоєння знань студентами виконується шляхом виконання ними завдань комп'ютерних практикумів та МКР. Таким чином, семестровий рейтинг студента з дисципліни складається з балів, котрі він отримує за:

- виконання та захист завдань комп'ютерних практикумів;
- виконання модульних контрольних робіт;
- відповіді на заліку (залікову роботу).

## 8.2 Система рейтингових (вагових) балів та критерії оцінювання

### Комп'ютерні практикуми

Вагові бали завдання кожного комп'ютерного практикуму наведено у таблиці 2.

Сумарний ваговий бал за даний контрольний захід складає 40 балів. Критерії оцінювання завдань включають якість їх виконання і якість захисту із оформленням звіту (таблиця 2).

Таблиця 2 – Вагові бали та критерії оцінювання лабораторних робіт (комп'ютерних практикумів )

Завдання	Критерії та бали	
	Виконання та захист зі звітом	Макс. сума балів
1	- завдання виконане та захищене без зауважень – 8 балів;	8
2	- завдання виконане достатньо повно з деякими похибками та захищене із зауваженнями – 5-7 балів;	8
3	- не повністю виконане завдання, при захисті частина відповідей відсутня або надано часткові відповіді – 2-4 бали;	8
4	- є суттєві зауваження без відповідей – 1 бал;	8
5		8
<b>Разом</b>		<b>40</b>

Виконаний та зарахований цикл всіх комп'ютерних практикумів є умовою допуску до семестрового контролю. Студенти, що на момент консультації перед семестровим контролем не захистили завдання комп'ютерних практикумів, не допускаються до основної здачі та готуються до перескладання.

Для допуску до перескладання семестрового контролю треба у визначений викладачем термін здати всі заборгованості з комп'ютерних практикумів.

### Модульна контрольна робота

Модульна контрольна робота складається з 10 тестових питань за темами, котрі були розглянуті при вивченні освітнього компоненту. Максимальний ваговий бал за даний контрольний захід для = 10 балів.

Оцінювання кожного питання в МКР здійснюється наступним чином:

- правильна відповідь на запитання: 1 бал;
- неправильна відповідь на запитання: 0 балів.

Виконана на позитивну оцінку МКР є умовою допуску до семестрового контролю. Студенти, що на момент консультації перед семестровим контролем не виконали модульну контрольну роботу, не допускаються до основної здачі та готуються до перескладання.

Для допуску до перескладання семестрового контролю треба у визначений викладачем термін виконати модульну контрольну роботу.

### 8.3 Календарний контроль

Календарний контроль провадиться двічі на семестр на 7 та 13 тижнях як моніторинг поточного стану виконання вимог силабусу - студент отримує «атестовано» під час першого та другого календарного контролю, якщо його поточний рейтинг складає не менше за 50% від максимальної кількості балів, можливої на момент контролю.

За результатами навчальної роботи за перші 7 тижнів «ідеальний студент» має набрати 18 балів. На першій атестації (7-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менший за 9 балів.

За результатами 13 тижнів навчання «ідеальний студент» має набрати 36 балів. На другій атестації (13-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менший за 18 балів.

### 8.4 Розрахунок шкали рейтингу

Семестровий контроль: залік.

У випадку централізованого рішення в університеті та з дозволу кафедри може здійснюватися перерахунок рейтингу, отриманого протягом семестру, за формулою, наведеною в Регламенті проведення семестрового контролю в дистанційному режимі.

Верхня межа рейтингової шкали з дисципліни становить 100 балів.

Максимальна сума вагових балів  $R1$  за виконання оцінювальних заходів протягом семестру становить 50 балів. Сума вагових балів, які отримує студент за роботу протягом семестру, обчислюється за формулою:

$$R1 = W + M$$

де  
 $W$  – сума балів, які студент отримав протягом семестру за оцінені завдання, перелічені в табл. 2;  
 $M$  – сума балів, отриманих за модульну контрольну роботу.

Залікова складова шкали  $R2 = 50$  балів.

Сумарний ваговий бал (максимальний) за семестр  $R$ , визначається як:

$$R = R1 + R2 = 40 + 10 + 50 = 100$$

Для отримання студентом відповідних оцінок його рейтингова оцінка  $R$  переводиться згідно з таблицею відповідності рейтингових балів оцінкам за університетською шкалою.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо

Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

### **9. Додаткова інформація з дисципліни (освітнього компонента)**

Якщо студент пред'являє сертифікат проходження курсів з інтернет-програмування, вебпрограмування, програмування мовами, орієнтованими на веб, у нього є можливість здати курс екстерном. Для цього студент проходить тестування та співбесіду з викладачем.

#### **Робочу програму навчальної дисципліни (силабус):**

**складено** старшим викладачем кафедри ІПІ Ковтунцем О.В.

**ухвалено** кафедрою ІПІ (протокол № 2/1 від 10.10.2025 р.)

**погоджено** Методичною комісією факультету (протокол № 3 від 17.10.2025 р.)