



**Міжнародний гуманітарний університет
Факультет іноземних мов, медіа та психології
Кафедра педагогіки та психології**

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
Основи дидактики

Галузь знань

01 Освіта / Педагогіка

Спеціальність

014.09 Середня освіта (Інформатика)

Назва освітньої програми

Інформатика та програмування

Рівень вищої освіти

перший (бакалаврський) рівень

Розробники і викладачі	Контактний тел.	E-mail
Доктор педагогічних наук, професор, практичний психолог, професор кафедри педагогіки та психології Єрмакова Світлана Станіславівна	+380982101888	ermakovasvitlana2011@ukr.net

1. АНОТАЦІЯ ДО КУРСУ

Метою вивчення навчальної дисципліни «Основи дидактики» є підготовка фахівців спеціальності 014.09 Середня освіта (Інформатика) за освітнім ступенем бакалавра відповідно до державних стандартів, встановлених освітньо-професійною програмою «Інформатика та програмування»; формування у майбутніх учителів системи теоретичних знань про загальні закономірності, принципи, методи та форми навчання, а також розвиток практичних умінь і навичок проектування освітнього процесу в закладах середньої освіти.

Основними **завданнями** вивчення дисципліни є:

- **теоретичні:** засвоєння фундаментальних понять дидактики (викладання, учіння, зміст освіти, методи контролю) та розуміння логіки навчального процесу;

- *практичні*: оволодіння методиками розробки поурочних планів, структурування навчального матеріалу з інформатики та програмування, вибору оптимальних методів навчання відповідно до вікових особливостей учнів;
- *технологічні*: формування здатності інтегрувати сучасні цифрові інструменти та інформаційно-комунікаційні технології в традиційну дидактичну систему для підвищення ефективності засвоєння знань;
- *прогностичні*: розвиток умінь аналізувати результати навчання, здійснювати діагностику знань учнів та коригувати освітню траєкторію.

Передумови для вивчення дисципліни. Перед вивченням навчальної дисципліни «*Основи дидактики*» здобувачі мають вивчати такі навчальні дисципліни, як Загальна педагогіка, Академічна доброчесність та професійна етика, Основи загальної психології.

2. ОЧІКУВАНІ КОМПЕТЕНТНОСТІ, ЯКІ ПЛАНУЄТЬСЯ СФОРМУВАТИ ТА ДОСЯГНЕННЯ ПРОГРАМНИХ РЕЗУЛЬТАТІВ

У процесі реалізації програми дисципліни «Основи дидактики» формуються наступні компетентності із передбачених освітньою програмою:

Інтегральна компетентність

ІК1. Здатність розв'язувати складні спеціалізовані задачі у сфері середньої освіти.

Загальні компетентності (ЗК)

ЗК9. Здатність виявляти ініціативу та підприємливість.

Фахові компетентності

ФК1. Здатність виконувати вимоги Державного стандарту базової середньої освіти у професійній діяльності з урахуванням предметної спеціальності, моделювати зміст навчання відповідно до обов'язкових результатів навчання учнів, планувати, організовувати освітній процес з використанням різних видів і форм навчально-пізнавальної діяльності, прогнозувати результати освітнього процесу.

ФК3. Здатність здійснювати інтегроване навчання учнів, добирати і використовувати сучасні та ефективні методики й технології навчання, виховання і розвитку учнів, використовувати інновації у професійній діяльності.

ФК5. Здатність здійснювати оцінювання та моніторинг результатів навчання учнів на засадах компетентнісного підходу, оцінювати рівень навчальних досягнень учнів, аналізувати результати їхнього навчання, навчати учнів самооцінювання та взаємооцінювання.

ФК17. Здатність здійснювати моніторинг власної педагогічної діяльності, навчатися впродовж життя.

Навчальна дисципліна «Основи дидактики» забезпечує досягнення **програмних результатів навчання (РН)**, передбачених освітньою програмою:

РН1. Знати та дотримуватися вимог Державного стандарту базової середньої освіти у професійній діяльності відповідно до предметної спеціальності; знати законодавчі вимоги щодо змісту загальної середньої освіти відповідного рівня та форм організації освітнього процесу (державні

стандарту, типові освітні програми, модельні навчальні програми); планувати освітній процес на основі освітньої програми закладу освіти і навчальних програм з предметів (інтегрованих курсів) з урахуванням вимог Державного стандарту базової середньої освіти.

РН6. Добирати дидактичні матеріали для вивчення учнями окремих тем/розділів навчальної програми відповідно до обов'язкових результатів навчання; добирати доцільні сучасні методики й технології навчання, виховання і розвитку учнів засобами освітньої галузі/навчального предмета (інтегрованого курсу) відповідно до визначених теми, мети і завдань уроку; використовувати навчальний матеріал з метою розвитку в учнів ключових компетентностей і вмінь, спільних для всіх компетентностей; навчати учнів застосовувати їх на практиці; моделювати навчальні заняття на основі компетентнісного, діяльнісного, особистісно зорієнтованого підходів.

РН9. Володіти методиками та інструментами оцінювання й моніторингу результатів навчання учнів, здійснювати різні види оцінювання результатів навчання учнів (формульвальне, поточне, підсумкове тощо) з використанням відповідних методик і критеріїв оцінювання, аналізувати результати навчання учнів з метою подальшого врахування їх в освітньому процесі; використовувати методи та прийоми розвитку в учнів здатності до самооцінювання та взаємооцінювання результатів навчання.

РН18. Організовувати навчальні заняття різних типів, застосовувати різні види й форми навчально-пізнавальної діяльності учнів.

РН19. Добирати та застосовувати інноваційні форми, методи, прийоми, засоби навчання у педагогічній діяльності, визначати власні освітні потреби та обирати відповідні види, форми, програми професійного розвитку.

Заплановані результати навчання за навчальною дисципліною у термінах НРК

Знання:

Зн1 Концептуальні наукові та практичні знання в області освіти та дидактики

Зн2 Критичне осмислення теорій, принципів, методів і понять у сфері дидактики та навчання

Уміння/Навички:

Ум1 Поглиблені когнітивні та практичні уміння/навички, майстерність та інноваційність на рівні, необхідному для розв'язання складних спеціалізованих задач і практичних проблем у сфері дидактики та навчання

Комунікація

K1 Донесення до фахівців і нефахівців інформації, ідей, проблем, рішень, власного досвіду та аргументації

K2 Збір, інтерпретація та застосування даних у сфері навчання

K3 Спілкування з професійних питань, у тому числі іноземною мовою, усно та письмово

Відповідальність і автономія

AB1 Управління складною професійною діяльністю чи проектами в області та навчання

AB2 Спроможність нести відповідальність за вироблення та ухвалення рішень у непередбачуваних навчальних контекстах

AB3 Формування суджень, що враховують соціальні, наукові та етичні аспекти

AB4 Організація та керівництво професійним розвитком осіб та груп

AB5 Здатність продовжувати навчання із значним ступенем автономії

3. ОБСЯГ ТА ОЗНАКИ КУРСУ

Загалом		Вид заняття (денне відділення / заочне відділення)			Ознаки курсу		
ЄКТС	годин	Лекційні заняття	Практичні заняття	Самостійна робота	Курс, (рік навчання)	Семестр	Обов'язкова / вибіркова
6	180	40 / 12	40 / 12	100 / 156	3	5	Обов'язкова

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин							
	Денна форма				Заочна форма			
	Усього	У тому числі			Усього	У тому числі		
		Лекц.	Практ.	Сам.		Лекц.	Практ.	Сам.
Тема 1. Дидактика як теорія освіти і навчання в цифрову епоху.	18	4	4	10	18	2	-	16
Тема 2. Процес навчання як цілісна система.	18	4	4	10	18	2	-	16
Тема 3. Закономірності та принципи навчання в ІТ-освіті.	18	4	4	10	18	2	2	14
Тема 4. Зміст освіти та державні стандарти навчання інформатики.	18	4	4	10	18	-	2	16
Тема 5. Методи навчання: від репродуктивних до дослідницьких.	18	4	4	10	18	-	2	16
Тема 6. Інноваційні технології навчання інформатики.	18	4	4	10	18	2	2	14
Тема 7. Форми організації навчання у комп'ютерному класі.	18	4	4	10	18	2	-	16
Тема 8. Дистанційне навчання та цифрові інструменти дидактики.	18	4	4	10	18	-	2	16
Тема 9. Діагностика та оцінювання навчальних досягнень учнів.	18	4	4	10	18	-	2	16
Тема 10. Проєктування та самоаналіз діяльності вчителя інформатики.	18	4	4	10	18	2	-	16
Усього годин	180	40	40	100	180	12	12	156
Підсумковий контроль – екзамен								

5. ТЕХНІЧНЕ Й ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ / ОБЛАДНАННЯ

Студенти отримують теми та питання курсу, основну і додаткову літературу, рекомендації, завдання та оцінки за їх виконання як традиційним шляхом, так і з використанням університетської платформи он-лайн навчання на базі Moodle. Окрім того, практичні навички у пошуку та аналізу інформації за курсом, з оформлення індивідуальних завдань, тощо, студенти отримують, користуючись університетськими комп'ютерними класами та бібліотекою.

6. САМОСТІЙНА РОБОТА

До самостійної роботи студентів щодо вивчення дисципліни Основи дидактики включаються:

1. Знайомство з науковою та навчальною літературою відповідно зазначених у програмі тем.
2. Опрацювання лекційного матеріалу.
3. Підготовка до практичних занять.
4. Консультації з викладачем протягом семестру.
5. Самостійне опрацювання окремих питань навчальної дисципліни.
6. Підготовка та виконання індивідуальних завдань у вигляді доповідей, презентацій.
7. Підготовка до підсумкового контролю.

Тематика та питання до самостійної підготовки та індивідуальних завдань

№ з/п	Назва теми	Кількість годин	
		денна форма	заочна форма
1	Тема 1. Дидактика як теорія освіти і навчання в цифрову епоху 1. Дидактика в цифрову епоху: предмет, завдання та категорії дидактики в контексті навчання програмування. 2. Трансформація об'єкта та предмета дидактики в умовах тотальної цифровізації освіти. 3. Дидактичний трикутник у програмуванні: взаємодія «Вчитель – Учень – Програмне середовище». 4. Еволюція категорій дидактики: від класичних «знань, умінь, навичок» до «цифрових компетентностей». 5. Етичні та дидактичні виклики: проблема академічної доброчесності при використанні готових рішень з мережі. Творче завдання Завдання спрямовані на переосмислення класичних категорій дидактики (викладання, навчання, зміст, методи) крізь призму цифрової трансформації освіти, що дозволяє майбутньому вчителю проектувати навчальний процес як високотехнологічну екосистему, яка відповідає Закону України «Про основні засади розвитку інформаційного суспільства»	10	14

та вимогам Цифрової компетентності вчителя, де об'єктом пізнання стає не лише синтаксис мов, а й алгоритмічна логіка взаємодії людини з інтелектуальними системами. 1. <i>Проект «Дидактичний реверс-інжиніринг»</i> <u>Завдання:</u> Оберіть популярний мобільний додаток для самостійного вивчення програмування (наприклад, <i>SoloLearn</i>, <i>Grasshopper</i> або <i>Mimo</i>). Проаналізуйте його «дидактичне ядро»: • Як у додатку реалізовано категорію «викладання» (через які інструменти подається контент)? • Які завдання дидактики вирішує цей застосунок краще, ніж традиційний підручник? • <u>Результат:</u> Складіть «Маніфест ідеального освітнього застосунку» з 5 пунктів, базуючись на виявлених перевагах та недоліках. 2. <i>Кейс «Дидактичний трикутник 2.0»</i> <u>Завдання:</u> Класичний дидактичний трикутник складається з вершин: <i>Вчитель</i> — <i>Учень</i> — <i>Зміст</i>. У цифрову епоху з'являється четвертий агент — Штучний інтелект (AI). • Намалюйте нову схему взаємодії («Дидактичний квадрат»). • Опишіть нові ролі: як змінюється категорія «учіння», якщо ШІ може миттєво згенерувати код? • <u>Результат:</u> Напишіть короткий діалог між Вчителем, Учнем та ШІ-помічником, де вони спільно вирішують проблему «вічного циклу» в коді, розподіляючи дидактичні ролі. 3. <i>Створення «Візуальної карти категорій»</i> <u>Завдання:</u> Використовуючи інструменти інтерактивного дизайну (Canva, Miro або Figma), створіть інфографіку «Дидактика програмування: від А до Я». • Візуалізуйте основні категорії (знання, вміння, навички) у вигляді компонентів комп'ютерної системи (наприклад: <i>Знання = База даних</i>, <i>Навички = Алгоритми в дії</i>, <i>Виховання = User Agreement</i>). • <u>Результат:</u> Плакат-інфографіка, що пояснює батькам учнів, чому вивчення коду — це не просто «натискання клавіш», а повноцінний освітній процес. • 4. <i>Дидактична дуель «Метод vs Технологія»</i> <u>Завдання:</u> Розділіть аркуш на дві частини. З одного боку опишіть класичний метод навчання (наприклад, пояснювально-ілюстративний). З іншого — його цифрову трансформацію (наприклад, інтерактивний відеоурок з вбудованим IDE). • Аргументуйте: чи змінюється предмет дидактики, коли «ілюстрація» стає динамічним симулятором? • <u>Результат:</u> Порівняльна таблиця-комікс, що демонструє еволюцію вчителя інформатики від 1990-х до 2026 року. 5. <i>Розробка «Smart-завдання» на межі дисциплін</i> <u>Завдання:</u> Дидактика цифрової епохи вимагає прагматичності змісту. Сконструуйте одне навчальне завдання з програмування, яке б одночасно вирішувало три завдання дидактики:		
---	--	--

	1. Освітнє: Засвоєння конструкції if-else. 2. Виховне: Формування культури поведінки в соцмережах (кібербулінг). 3. Розвивальне: Розвиток емпатії або критичного мислення. • <u>Результат</u> : Картка-завдання «Напиши чат-бота — цифрового етичного радника».		
2	Тема 2. Процес навчання як цілісна система. - Психолого-дидактичні основи навчання інформатики: формування алгоритмічного мислення в учнів різних вікових груп. - Когнітивні особливості формування алгоритмічного мислення в учнів початкової школи (період конкретних операцій за Піаже). - Сенситивні періоди для переходу від візуально-блочного (Scratch) до текстового (Python) програмування. - Розвиток абстрактного мислення під час вивчення об'єктно-орієнтованої парадигми в підлітковому віці. - Психологія «страху перед помилкою» в коді та методи дидактичної підтримки учня. Творче завдання Завдання спрямовані на опанування методик адаптації складних логічних конструкцій до вікових когнітивних особливостей учнів (згідно з теоріями Ж. Піаже та Л. Виготського), що забезпечує реалізацію принципу наступності між початковою, базовою та старшою школою відповідно до вимог Державного стандарту, а також на вироблення вміння проєктувати завдання, що розвивають операційні компоненти мислення (аналіз, синтез, абстрагування) через візуальне, блочне та текстове програмування 1. Проєкт «Алгоритми без екрана» (CS Unplugged для молодшої школи) Контекст: У молодшому шкільному віці (6–9 років) переважає наочно-образне мислення. Дитині важливо «помацати» алгоритм, перш ніж писати його в коді. Завдання: Розробити настільну гру або рухливий квест «Живий робот», який навчає базовим алгоритмічним конструкціям (лінійність, цикл, розгалуження) без використання комп'ютера. Вимоги: - Створити набір фізичних карток-команд (вперед, ліворуч, стрибок, повтори 3 рази). - Прописати сюжет (наприклад, «Допоможи бджілці дістатися квітки, оминаючи павутиння»). - Обґрунтувати, як ця активність розвиває вміння декомпонувати велику задачу на маленькі кроки. Результат: Методичний опис гри з фотографіями або ескізами карток та полем. 2. Кейс «Переклад зі студентської на дитячу» (Дидактична адаптація) Контекст: Для учнів середньої ланки (10–13 років) характерний перехід до абстрактного мислення, але їм все ще потрібні яскраві метафори для розуміння складних структур. Завдання: Обрати складне алгоритмічне поняття (наприклад, «Цикл з умовою (while)» або «Рекурсія») і пояснити його трьома різним аудиторіям за допомогою метафор, не використовуючи технічну термінологію. Цільові групи:	10	14

	1. Першокласники (пояснення через казку або ігрову дію). 2. П'ятикласники (через побутовий приклад або спортивну вправу). 3. Одинадятикласники-гуманітарії (через соціальну ситуацію або літературний сюжет). • <u>Результат</u>: Порівняльна таблиця «Поняття — Метафора — Психологічне обґрунтування вибору». 3. Дидактична лабораторія «Пастка для алгоритміста» (Аналіз помилок) <u>Контекст</u>: У підлітковому віці (14–16 років) важливо розвивати критичне та аналітичне мислення. Найкращий спосіб це зробити — навчити аналізувати чужі та власні помилки. <u>Завдання</u>: Створити «Дидактичний посібник шкідливих алгоритмів». Потрібно написати три короткі фрагменти коду (наприклад, на Python або Scratch), які містять логічні (не синтаксичні!) помилки, що призводять до неочікуваних результатів. • <u>Вимоги</u>: ○ Помилка 1: Призводить до вічного циклу. ○ Помилка 2: Призводить до того, що умова ніколи не виконується. ○ Помилка 3: Дає неправильний результат через порушення пріоритетності дій. ○ Додатково: Скласти чек-лист для учня «5 запитань до свого алгоритму», який допоможе їм самостійно знаходити ці помилки. • <u>Результат</u>: Картки із завданнями «Знайди баг» та коротке есе про те, чому дидактично корисніше дати учню зламаний алгоритм, ніж ідеально правильний.		
3	Тема 3. Закономірності та принципи навчання в ІТ-освіті 1. Зміст ІТ-освіти: аналіз модельних навчальних програм з інформатики та вибір стеків технологій для школи. 2. Дидактичні критерії вибору першої мови програмування для шкільного курсу (порівняльний аналіз Python, JavaScript, C++). 3. Структура модельних програм НУШ: баланс між теоретичною інформатикою та прикладним програмуванням. 4. Проблема актуальності змісту: методичне оновлення навчального матеріалу відповідно до темпів розвитку ІТ-індустрії. 5. Варіативні складники змісту: дидактичні засади впровадження курсів за вибором (Cybersecurity, GameDev, Data Science). Творче завдання Завдання спрямовані на розвиток аналітичних навичок проектування змісту навчання, що базуються на критичному аналізі модельних навчальних програм та їх відповідності Державному стандарту базової середньої освіти, а також на вироблення здатності дидактично обґрунтовано обирати актуальні стеки технологій та інструментарій розробки, реалізуючи право вчителя на академічну свободу та автономію у виборі засобів навчання (згідно зі ст. 54 Закону України «Про освіту») 1. <i>Творча майстерня «Конструктор навчального стеку»</i>	10	16

	Контекст: Державний стандарт визначає <i>результати</i> навчання (що учень має вміти), але не обмежує вчителя у виборі конкретних інструментів (мов програмування, софту). Завдання: Спроектувати «Ідеальний технологічний пакет» для вивчення програмування у 7–9 класах на основі однієї з чинних модельних програм. • Вимоги: ○ Вибрати базову мову програмування та обґрунтувати її дидактичну доцільність (наприклад, чому Python, а не C++). ○ Підібрати допоміжні інструменти (IDE, системи контролю версій, візуалізатори). ○ Скласти «Матрицю відповідності»: показати, як обраний стек технологій дозволяє досягти конкретних очікуваних результатів, прописаних у Державному стандарті. • Результат: Презентація-захист технологічного стеку «School IT Stack 2026». 2. Дидактична експертиза «Програма vs Реальність» Контекст: Програми оновлюються раз на кілька років, а бібліотеки та фреймворки — щомісяця. Вчитель має вміти адаптувати контент, не порушуючи законодавчу базу. Завдання: Провести «краш-тест» одного модуля чинної навчальної програми (наприклад, «Веб-технології» або «Бази даних»). • Вимоги: ○ Знайти в програмі теми, які за останні 2-3 роки стали неактуальними (наприклад, застарілі теги HTML чи формати файлів). ○ Запропонувати сучасну альтернативу, яка б відповідала змістовій лінії програми. ○ Підготувати коротку пояснювальну записку для адміністрації школи, чому така заміна є методично виправданою та законною (посилання на автономію вчителя згідно із Законом «Про освіту»). • Результат: Проєкт оновленого тематичного планування з коментарями. 3. Аналітичний батл «Лінійна / Концентрична модель» Контекст: Побудова змісту ІТ-освіти може бути лінійною (вчимо теми послідовно один раз) або концентричною (повертаємося до тем кожного року, поглиблюючи їх). Завдання: Створити порівняльну інфографіку або інтерактивну карту змісту для вивчення теми «Алгоритмізація та програмування» з 5 по 11 клас за двома різними підходами. • Вимоги: ○ Візуалізувати шлях учня: які поняття додаються на кожному етапі («концентрі»). ○ Визначити критичні точки «перевантаження» змісту (де програма стає занадто складною для віку учня). ○ Висловити аргументовану позицію: яка модель побудови змісту є ефективнішою для формування навичок професійного програмування. • Результат: Інфографіка-порівняння (наприклад, у Miro або Canva) та коротке есе-обґрунтування.		
4	Тема 4. Зміст освіти та державні стандарти навчання інформатики 1. Закономірності та принципи навчання: реалізація принципів науковості та наочності при візуалізації абстрактних структур даних.	10	16

2. Принцип науковості: як коректно адаптувати складні алгоритми (сортування, пошук) для шкільного рівня. 3. Когнітивна теорія мультимедійного навчання (Р. Майєр) при візуалізації структур даних. 4. Дидактичні можливості статичної та динамічної наочності у викладанні архітектури комп'ютера. 5. Принцип міцності знань: роль рефакторингу та повторного використання коду в закріпленні навичок. Творче завдання Завдання спрямовані на розвиток методичної майстерності у трансформації складних теоретичних концепцій програмування у візуальні дидактичні образи, що забезпечує дотримання принципу науковості (згідно з вимогами Державного стандарту щодо точності термінології) та принципу наочності, а також сприяє запобіганню когнітивного перевантаження учнів при роботі з абстракціями, відповідно до психолого-педагогічних вимог щодо цифровізації змісту освіти. <i>Завдання 1. Проєктування динамічної моделі «Абстракція vs Реальність»</i> Мета: Навчитися реалізовувати принцип наочності через створення інтерактивних візуалізацій, що не суперечать принципу науковості. <u>Зміст завдання:</u> Оберіть одну складну структуру даних (наприклад, <i>бінарне дерево пошуку</i> або <i>зв'язний список</i>). Розробіть сценарій інтерактивної візуалізації для пояснення цієї теми учням 9-го класу. • Теоретична частина: Визначте 3 ключові наукові властивості обраної структури, які обов'язково мають бути відображені (наприклад, для дерева — ієрархічність, наявність кореня, принцип розгалуження). • <i>Результат:</i> 1) Створіть прототип візуалізації (це може бути набір анімованих слайдів, посилання на створену модель у <i>Scratch</i> або використання інструменту <i>Python Tutor</i>). 2) Дидактичний аналіз: Обґрунтуйте, як ваша наочність допомагає уникнути «наївних уявлень» та підтримує наукову точність визначень. <i>Завдання 2. Дидактична експертиза візуальних образів</i> Мета: Розвиток критичного мислення щодо використання метафор та ілюстрацій у підручниках з програмування. <u>Зміст завдання:</u> Знайдіть у відкритих джерелах (підручники, онлайн-курси, YouTube) 3 приклади візуалізації теми «Стек та Черга». • Аналітична частина: Проаналізуйте кожен приклад за критеріями: 1. Чи не спотворює метафора (наприклад, «стопка тарілок» для стека) наукову сутність операцій LIFO та FIFO? 2. Чи відповідає рівень деталізації візуалізації принципу доступності для певної вікової групи? • <i>Результат:</i> Запропонуйте власну «удосконаленну» візуалізацію, яка виправляє виявлені недоліки (наприклад, додайте візуальне відображення покажчиків Top або Head/Tail, щоб наочність стала більш науковою). <i>Завдання 3. Лабораторія «Когнітивна візуалізація коду»</i>		
--	--	--

	Мета: Навчитися використовувати інструменти автоматичної візуалізації для реалізації принципу свідомості та активності учнів. <u>Зміст завдання:</u> Підготуйте фрагмент практичної роботи для теми «Рекурсія на прикладі обчислення факторіалу або чисел Фібоначчі». - Практична частина: Використовуючи сервіс візуалізації виконання коду (наприклад, cscircles.pyschool.net або pythontutor.com), складіть покрокову інструкцію для учня. - Методична частина: Сформулюйте 5 контрольних запитань, які змушують учня аналізувати «стек викликів» на екрані (наприклад: «На якому кроці візуалізації змінна <i>n</i> набуває значення 0? Що відбувається з пам'яттю в цей момент?»). - Рефлексія: Поясніть, як використання таких інструментів забезпечує перехід від наочного сприйняття до глибокого наукового розуміння алгоритму.		
5	Тема 5. Методи навчання: від репродуктивних до дослідницьких - Методи навчання в ІТ: від репродуктивних методів до парного програмування та методу «Peer-to-peer». - Методика «Pair Programming» (парне програмування): дидактичні переваги та сценарії реалізації в класі. - Метод «Peer-to-peer» (рівний-рівному): організація взаємонавчання під час групових ІТ-проектів. - Репродуктивне навчання: коли використання методу «слідуй за вчителем» (Live Coding) є виправданим. - Дослідницькі методи: використання технік Reverse Engineering (зворотного проектування) для аналізу готового коду. Творче завдання Завдання спрямовані на опанування інструментарію інтерактивного навчання та командної взаємодії, що відповідає вимогам Професійного стандарту вчителя щодо формування соціальної та громадянської компетентностей учнів, а також положенням Державного стандарту базової середньої освіти в частині розвитку навичок комунікації, спільної розробки складних цифрових продуктів та впровадження методик взаємонавчання (Peer-to-peer) як засобу забезпечення активної позиції учня в освітньому процесі. - Методичний конструктор «Сходінки майстерності» Контекст: Навчання програмуванню — це шлях від копіювання коду за вчителем до створення власних систем. Вчитель має вміти комбінувати методи для поступового зростання учня. <u>Завдання:</u> Розробити дидактичний план вивчення теми «Функції в Python (або C++)», використовуючи три різні рівні методів. - Рівень 1 (Репродуктивний): Створити вправу «Live Coding» (слідуй за вчителем), де учні відтворюють базову структуру функції. - Рівень 2 (Частково-пошуковий): Розробити завдання на «рефакторинг», де учням дається готовий лінійний код, який вони мають самостійно перетворити на набір функцій. - Рівень 3 (Дослідницький): Сформулювати міні-проект для парного програмування, де одна дитина виступає «штурманом» (проектує логіку), а інша — «водієм» (пише код).	10	16

	• <u>Результат</u>: Порівняльна таблиця активності вчителя та учнів на кожному етапі. 2. <i>Сценарій дидактичної гри «Peer-to-Peer Code Review»</i> Контекст: Метод «рівний-рівному» (P2P) — один із найефективніших в ІТ, оскільки він розвиває навички читання чужого коду та конструктивної критики. Завдання: Розробити регламент і робочі матеріали для проведення уроку-взаємоперевірки (Code Review) для учнів 9-10 класів. • Вимоги до завдання: 1. Підготувати чек-лист для учня-рецензента (на що звертати увагу: назви змінних, відступи, коментарі, працездатність). 2. Сформулювати «правила етичної критики» для учнів (як сказати про помилку, не образивши однокласника). 3. Описати систему дидактичного оцінювання: як вчителю оцінити не тільки автора коду, а й якість рецензії, наданої «піром». • <u>Результат</u>: Пакет інструкцій «Peer-Review інструментарій для вчителя». 3. <i>Творча рефлексія «Методична інверсія»</i> Контекст: Іноді найкращий спосіб навчити — це поміняти ролі або змодельовати ситуацію, де традиційний метод не працює. Завдання: Створити відеовізитку або інтерактивний плакат (у Canva/Genially), що демонструє переваги активних методів над догматичними. • Зміст: ○ Показати «анти-приклад»: як виглядає нудне репродуктивне навчання програмуванню (читання підручника вголос). ○ Протиставити йому метод «Евристичної бесіди» або «Навчання через відкриття» (Inquiry-based learning). ○ Обґрунтувати, чому для підготовки майбутніх програмістів методи P2P та Agile-ігри є більш значущими, ніж механічне зазубрювання синтаксису. • <u>Результат</u>: Мультимедійний продукт «Чому програмуванню не можна навчити лекцією?».		
6	Тема 6. Інноваційні технології навчання інформатики 1. Проблемне навчання на уроках інформатики: дидактика дебагінгу (debugging) та розв’язання олімпіадних задач. 2. Методика «Debug-First»: навчання через пошук та виправлення помилок у навмисно пошкодженому коді. 3. Етапи розв’язання олімпіадних задач з точки зору теорії проблемного навчання (аналіз, формалізація, алгоритмізація). 4. Розвиток критичного мислення через порівняння ефективності різних алгоритмів (Big O notation для школи). 5. Метод евристичних запитань під час консультації учнів над складними програмними проектами.	10	16

Творче завдання Завдання спрямовані на розвиток методичної компетентності вчителя щодо створення проблемних ситуацій, які стимулюють аналітичне мислення учнів, та впровадження технік керованого пошуку помилок у межах дотримання принципів академічної доброчесності (згідно зі ст. 42 Закону України «Про освіту») та вимог Державного стандарту щодо формування здатності учнів критично оцінювати та вдосконалювати власні програмні рішення. 1. Методичний конструктор «Дидактична пастка» Контекст: Навмисне внесення помилок у код — це потужний метод проблемного навчання. Вчитель має вміти створювати такі "пастки", які б підштовхували учня до глибокого розуміння логіки алгоритму. Завдання: Розробити набір із 3-х дидактичних карток до теми «Цикли з умовами». - Картка №1 «Синтаксичний лабіринт»: Код містить дрібні помилки синтаксису, які легко знайти компілятору, але важливо побачити оку. - Картка №2 «Логічна прірва»: Код успішно запускається, але видає неправильний результат (наприклад, нескінченний цикл або помилка "на одиницю" — <i>off-by-one error</i>). - Картка №3 «Ефективний вибір»: Код працює правильно, але він катастрофічно неоптимальний (наприклад, $O(n^2)$ замість $O(n)$). Учень має "відбачити" не працездатність, а швидкість. Результат: Картки та методичні вказівки для вчителя: які запитання ставити учневі, щоб він сам знайшов проблему. 2. Кейс-стаді «Анатомія олімпіадної задачі» Контекст: Олімпіадні задачі часто здаються учням нездоланими. Дидактика проблемного навчання пропонує метод декомпозиції та поетапного відкриття. Завдання: Обрати одну олімпіадну задачу середнього рівня складності (наприклад, з платформи <i>e-olymp</i> або <i>LeetCode</i>). - Етап декомпозиції: Розбити розв'язання задачі на 3 підпроблеми (наприклад: 1. Зчитування та структурування даних; 2. Побудова математичної моделі; 3. Реалізація та оптимізація). - Етап "Scaffolding" (Підтримка): Сформулювати 3 підказки для учнів. Кожна наступна підказка має бути більш конкретною, ніж попередня, але не давати готового коду. Результат: Сценарій розбору задачі за методом "навчання через відкриття". 3. Творчий проєкт «Щоденник дебаггера» (Інтерактивний гайд) Контекст: Дебаггінг — це не просто виправлення коду, це певна культура мислення та використання інструментів. Завдання: Створити мультимедійну пам'ятку для учнів «Мистецтво полювання на багів». - Зміст: - Візуалізація методів: пояснити методи "гумової качечки", "трасування вручну" та "ділення коду навпіл" (бінарний пошук помилки). - Інструкція до інструментів: як використовувати <i>Breakpoints</i> та <i>Watch variables</i> у сучасних IDE. - Психологічний аспект: поради, як зберігати спокій, коли програма не працює, і як розвинути "детективний азарт".		
---	--	--

	• <u>Результат</u>: Лонгвід, інфографіка або коротке відео, яке вчитель може використовувати як вступ до теми.		
7	Тема 7. Форми організації навчання у комп'ютерному класі 1. Форми організації навчання: традиційний урок VS хакатон, воркшоп та лабораторна робота. 2. Дидактичний потенціал шкільного Хакатону: організація інтенсивної командної розробки продукту. 3. Специфіка Воркшопу (Workshop): як організувати швидке оволодіння новою бібліотекою чи фреймворком. 4. Трансформація кабінету інформатики: від традиційного класу до гнучкого IT-хабу (MakerSpace). 5. Лабораторна робота в програмуванні: структура інструктажу та критерії успішності виконання. Творче завдання Завдання спрямовані на опанування навичок диференціації та гнучкого моделювання форм освітньої діяльності, що дозволяють вчителю ефективно реалізувати діяльнісний підхід Нової української школи (НУШ) та принципи педагогічної автономії (згідно зі ст. 54 Закону України «Про освіту»), балансуючи між академічними вимогами та інтенсивною командною розробкою IT-продуктів у межах безпечного освітнього середовища (відповідно до Санітарного регламенту для закладів загальної середньої освіти). 1. <i>Дидактичний батл «Урок чи Хакатон: Обираємо формат»</i> Контекст: Не кожен тему можна викласти у форматі хакатону, і не кожен урок дозволяє створити реальний продукт. Вчитель має бачити дидактичну доцільність кожної форми. <u>Завдання</u>: Оберіть одну велику тему з програмування (наприклад, «Розробка вебсайту на HTML/CSS» або «Створення Telegram-бота»). Складіть дві альтернативні дорожні карти вивчення цієї теми: • Варіант А (Традиційний): Серія з 4-х уроків (вивчення теорії, тренувальні вправи, контрольна робота). • Варіант Б (Інноваційний): Короткий вступний воркшоп + 6-годинний міні-хакатон. • Аналітична частина: Порівняйте ці формати за критеріями: обсяг засвоєних теоретичних знань, якість практичного продукту та рівень емоційного залучення учнів. • <u>Результат</u>: Порівняльна таблиця-обґрунтування «Ефективність форматів для теми Х». 2. <i>Проектування «Workshop-сценарію» для шкільної IT-студії</i> Контекст: Воркшоп передбачає інтенсивне навчання під керівництвом майстра, де теорія подається порційно і відразу впроваджується в дію. <u>Завдання</u>: Розробити детальний план 90-хвилинного воркшопу на тему «Перші кроки в GameDev: створюємо гру на PyGame (або Scratch)». • Вимоги до структури: 1. Micro-teaching: Як подати теорію за 10 хвилин? 2. Scaffolding: Підготувати «скелет» коду, який учні будуть доповнювати. 3. Етап «Майстер-класу»: Описати послідовність дій, які учні повторюють за вчителем синхронно.	10	16

	4. Творчий блок: Сформулювати 3 варіанти «фіч» (додаткових функцій), які учні мають додати до гри самостійно наприкінці заняття. <u>Результат:</u> Технологічна карта воркшопу з додатками (заготовки коду, чек-листи). 3. <i>Творча робота 2.0: «Дидактичний дизайн інструкції»</i> Контекст: Традиційні лабораторні роботи часто перетворюються на механічне копіювання тексту з методички. Сучасна лабораторна має стимулювати мислення. Завдання: Переробити стандартну лабораторну практичну роботу (наприклад, «Робота з масивами») у формат «Дослідницької лабораторії». - Творчий підхід: - Замість покрокової інструкції «напишіть цей код», дайте завдання «дослідіть, як зміниться швидкість сортування при збільшенні масиву в 100 разів». - Додайте блок «Що пішло не так?»: фрагмент коду з помилкою, яку треба виправити, щоб завершити роботу. - Включіть етап візуалізації результатів (побудова графіка або діаграми на основі отриманих даних). <u>Результат:</u> Оновлений бланк лабораторної роботи, який спонукає учня до експерименту, а не простого відтворення.		
8	Тема 8. Дистанційне навчання та цифрові інструменти дидактики - Дистанційне та змішане навчання (Blended Learning): організація самостійної роботи учнів у LMS (Moodle, Google Classroom). - Проектування навчальної траєкторії в LMS: логіка розбиття курсу програмування на мікро-модулі. - Дидактичні інструменти підтримки самостійної роботи: використання інтерактивних відеоінструкцій та форумів. - Модель «Rotational Lab» (ротація станцій) у кабінеті інформатики при обмеженій кількості ПК. - Синхронна та асинхронна комунікація: як забезпечити ефективний фідбек у дистанційному режимі. Творче завдання Завдання спрямовані на те, вироблення навички проектування цілісного навчального середовища, яке відповідає державним вимогам щодо дистанційної освіти (зокрема, Наказу МОН №1115 «Положення про дистанційну форму здобуття повної загальної середньої освіти»). <i>Проект-моделювання «Цифровий кабінет за стандартом»</i> Контекст: Згідно із законодавством, дистанційне навчання має забезпечувати регулярну взаємодію (синхронний та асинхронний режими) та облік навчальних досягнень. <u>Завдання:</u> Створити структуру та наповнення одного тематичного модуля (наприклад, «Основи вебдизайну») у Google Classroom або Moodle, дотримуючись «дидактичного мінімуму». - Вимоги до структури: - Інформаційний блок: Відеоінструкція (власна або підібрана) + короткий конспект. - Діяльнісний блок: Практичне завдання з чіткими критеріями оцінювання, прикріпленими у системі.	10	16

	3. Блок зворотного зв'язку: Створення посилання на синхронну зустріч (Meet/Zoom) та форуму/чату для асинхронних консультацій. 4. Контрольний блок: Автоматизований тест із налаштованим дедлайном та обмеженням часу. Результат: Скриншоти розгорнутого курсу або гостьовий доступ для викладача. 2. <i>Дидактичний конструктор «Ротація станцій у коді»</i> Контекст: Змішане навчання (Blended Learning) дозволяє ефективно використовувати обмежену кількість комп'ютерів у класі. Найпопулярніша модель — «Ротація станцій». Завдання: Розробити план-схему уроку інформатики для 8-го класу за моделлю «Ротація станцій». - Вимоги до станцій: - Станція «Робота з вчителем»: Пояснення складної логіки (наприклад, вкладені цикли). - Станція «Онлайн-навчання»: Самостійна робота в LMS (інтерактивна вправа на Stepik або Code.org). - Станція «Проектна група»: Командна робота над дебагінгом коду або створенням алгоритму на папері (CS Unplugged). - Законодавчий аспект: Врахувати Санітарний регламент щодо безперервної роботи за екраном (не більше 20-25 хв для цієї вікової групи). Результат: Графічна схема уроку з описом завдань на кожній станції та таймінгом переміщення. 3. <i>Аналітична записка «Академічна доброчесність у хмарі»</i> Контекст: Стаття 42 Закону України «Про освіту» вимагає дотримання академічної доброчесності. У дистанційному навчанні програмування це стає викликом через легкість копіювання коду. Завдання: Розробити методичну інструкцію для вчителів «Як запобігти плагіату коду при дистанційному навчанні». - Творчий пошук: - Запропонувати 3 варіанти «завдань, що не списуються» (наприклад, індивідуалізація проєкту через додавання особистих даних або відеопояснення логіки свого коду учнем). - Описати процедуру захисту роботи в синхронному режимі (бліц-опитування за рядками коду). - Скласти пам'ятку для учнів про те, де проходить межа між «допомогою Stack Overflow» та «плагіатом». Результат: Текстовий документ або інтерактивний плакат для учнів та колег.		
9	Тема 9. Діагностика та оцінювання навчальних досягнень учнів - Методика оцінювання в програмуванні: критерії якості коду, автоматизоване тестування та формувальне оцінювання проєктів. - Автоматизоване тестування (Auto-grading): дидактичні переваги та обмеження використання платформ (напр. E-olymp, LeetCode). - Формувальне оцінювання: використання «карт прогресу» та коммітів (GitHub) для відстеження росту учня. - Критеріальна матриця (Rubric): розробка параметрів оцінювання чистоти коду та архітектурних рішень. - Технологія Peer-assessment: залучення учнів до оцінювання робіт однокласників (Code Review).	10	16

	Творче завдання Завдання допоможуть опанувати сучасні підходи до оцінювання в ІТ-освіті, спираючись на Наказ МОН №1093 (щодо оцінювання результатів навчання учнів у НУШ) та критерії професійної оцінки програмного забезпечення. 1. <i>Проект «Створення критеріальної матриці (Rubric) для коду»</i> Контекст: Оцінювання в програмуванні не може обмежуватися лише результатом «працює/не працює». Згідно з концепцією НУШ, оцінювання має бути прозорим, а критерії — відомими учневі заздалегідь. <u>Завдання:</u> Розробити детальну рубрику для оцінювання підсумкового проєкту (наприклад, «Гра на Python» або «Веб-сайт»). - Вимоги до розробки: Розділити оцінку на 4 рівні (Початковий, Середній, Достатній, Високий) за такими критеріями: - Функціональність: Чи виконує програма завдання? (Законодавча база: досягнення очікуваних результатів навчання). - Якість та архітектура коду: Назви змінних, відсутності дублювання (DRY), модульність. - Документування: Наявність коментарів та опис логіки. - Творчий підхід: Оригінальність алгоритмічних рішень. <u>Результат:</u> Таблиця-матриця оцінювання, готова до використання на уроці. 2. <i>Кейс-стаді «Дидактика автоматизованого тестування»</i> Контекст: У професійному програмуванні код перевіряють тести (Unit tests). У школі це реалізується через онлайн-платформи (E-olymp, LeetCode). Вчитель має розуміти, як результати тестів транслювати в оцінку за державним стандартом. <u>Завдання:</u> Створити методичний кейс «Від тестів до балів». - Зміст завдання: - Оберіть складну задачу з програмування та пропишіть для неї 5 тестових сценаріїв (наприклад: порожні вхідні дані, максимально великі числа, некоректні типи). - Обґрунтуйте: скільки балів отримає учень, якщо пройшло 3 з 5 тестів? Яку якісну характеристику ви надасте такій роботі згідно з формувальним оцінюванням? - Запропонуйте інструкцію для учня: як інтерпретувати помилку «Wrong Answer» або «Time Limit Exceeded» для самостійного покращення коду. <u>Результат:</u> Картка завдання з описом тестів та шкалою перерахунку у шкільну оцінку. 3. <i>Лабораторія «Щоденник прогресу: Формувальне оцінювання»</i> Контекст: Формувальне оцінювання за законодавством спрямоване на відстеження вступу учня, а не на констатацію результату. У програмуванні це найкраще робити через ітерації та Code Review. <u>Завдання:</u> Розробити шаблон «Картки зворотного зв'язку» для тривалого навчального проєкту. - Складові творчого завдання:		
--	---	--	--

	1. Self-assessment блок: Запитання до учня («Що було найскладнішим в алгоритмі?», «Що я навчився робити самостійно?»). 2. Peer-assessment блок: Місце для відгуку однокласника про читабельність коду (Методика «Дві зірки і побажання»). 3. Teacher-feedback блок: Поради вчителя щодо оптимізації, а не просто виправлення помилок. • <u>Результат</u>: Дизайн «Дидактичного трека прогресу» (у Google Sheets, Notion або у формі паперового буклета).		
10	Тема 10. Проектування та самоаналіз діяльності вчителя інформатики 1. Педагогічний дизайн уроку інформатики: проектування технологічної карти сучасного уроку з використанням ШІ-інструментів. 2. AI-driven Instructional Design: використання ШІ для генерації різнорівневих завдань та персоналізації контенту. 3. Проектування технологічної карти уроку за моделлю ADDIE або ASSURE в умовах ІТ-освіти. 4. Дидактична експертиза ШІ-контенту: як вчителю верифікувати згенеровані алгоритми та пояснення. 5. Методика створення запитів (Prompt Engineering) для підготовки дидактичних матеріалів до уроку. Творче завдання Завдання спрямовані на оволодіння навичками педагогічного дизайну (Instructional Design) з використанням генеративного штучного інтелекту, спираючись на Професійний стандарт вчителя та вимоги НУШ щодо цифровізації освітнього процесу. <u>1. Воркшоп-проект «AI-Co-pilot для вчителя: від ідеї до структури»</u> Контекст: Професійний стандарт вчителя передбачає здатність педагога використовувати цифрові технології для планування навчання. ШІ може стати асистентом, який допомагає подолати «страх чистого аркуша». <u>Завдання</u>: Створити технологічну карту уроку на тему «Алгоритми штучного інтелекту в нашому житті», використовуючи ШІ (ChatGPT, Claude або Gemini) як методичного консультанта. • Вимоги до процесу: 1. Сформулювати серію промптів (запитів) для ШІ: від визначення очікуваних результатів (згідно з Держстандартом) до генерації ідей для зацікавлення учнів (hook). 2. Проаналізувати відповіді ШІ та внести корективи: де ШІ запропонував занадто складні завдання, а де — порушив логіку дидактики. • <u>Результат</u>: Портфоліо промптів та фінальна технологічна карта уроку, де чітко позначено блоки, розроблені спільно з ШІ. <u>2. Дидактична експертиза «ШІ-генерація навчальних матеріалів»</u> Контекст: Законодавство України (Закон «Про авторське право і суміжні права») та принципи академічної доброчесності вимагають від вчителя критичного ставлення до контенту, згенерованого не людиною.	10	16

	<u>Завдання:</u> Розробити дидактичне забезпечення до уроку програмування (тести, приклади коду, практична задача), згенероване ІІІ, та провести його експертизу. - Складові творчого завдання: - Згенерувати за допомогою ІІІ три варіанти коду до теми «Списки в Python» (правильний, з помилкою, неоптимальний). - Провести «Верифікацію якості»: перевірити код на працездатність та відповідність принципу доступності для школярів. - Розробити чек-лист «Етичне використання ІІІ на уроці інформатики» для учнів, базуючись на принципах академічної доброчесності. - Результат: Пакет дидактичних матеріалів з поміткою «Verified by Teacher» та аналітичний коментар щодо ризиків використання ІІІ-коду. <i>3. Редизайн уроку за моделлю «Smart-планування»</i> Контекст: Сучасний урок має бути гнучким. Використання ІІІ дозволяє вчителю за лічені хвилини адаптувати одну й ту саму технологічну карту під різні рівні класу (інклюзія, обдаровані учні, базовий рівень). <u>Завдання:</u> Взяти за основу класичний план-конспект уроку (наприклад, «Типи даних») і за допомогою ІІІ трансформувати його в індивідуалізовану технологічну карту. - Вимоги до редизайну: - Диференціація: ІІІ має згенерувати три варіанти практичних завдань до однієї теми (для візуалів, аудіалів та кінестетиків або за рівнями складності). - Таймінг-менеджмент: Використати ІІІ для оптимізації розподілу часу на уроці (враховуючи обов'язкові фізкультхвилинки згідно з Санітарним регламентом). - Інтерактивність: Запросити у ІІІ ідею для «цифрової рефлексії» наприкінці уроку (використання Mentimeter, Kahoot тощо). - Результат: Гнучка технологічна карта (матриця), яка показує варіативність шляхів навчання для різних учнів в межах одного уроку.		
	Усього годин	100	156

7. ВИДИ ТА МЕТОДИ КОНТРОЛЮ

Види контролю	Складові оцінювання	
	Для іспиту	Для заліку
поточний контроль , який здійснюється у ході: проведення практичних (лабораторних) занять, виконання самостійної роботи, індивідуальних завдань, дослідна робота здобувача тощо.	50%	100%
підсумковий контроль	50%	

Методи діагностики знань (контролю)	фронтальне опитування; наукова доповідь, есе, усне повідомлення, індивідуальне опитування; робота у групах; ділова гра, розв'язання ситуаційних завдань, кейсів, практичних завдань, екзамен.
--	---

Питання до екзамену

1. Дидактика як наука про навчання: предмет, завдання та її значення для вчителя інформатики.
2. Основні категорії дидактики: освіта, навчання, викладання, учіння, знання, вміння, навички.
3. Процес навчання як цілісна система: компоненти (цільовий, стимулюючо-мотиваційний, змістовий, операційно-діяльнісний, контрольно-регулювальний, оцінно-результативний).
4. Взаємозв'язок викладання та учіння в умовах комп'ютеризованого навчання.
5. Реалізація освітньої, виховної та розвивальної функцій навчання при вивченні мов програмування.
6. Психолого-педагогічні особливості пізнавальної діяльності учнів під час роботи з цифровим контентом.
7. Закономірності навчання: зовнішні (соціальні) та внутрішні (педагогічні).
8. Принцип науковості та його реалізація через актуалізацію стеку технологій у шкільній програмі.
9. Принцип систематичності та послідовності: від візуального програмування (Scratch) до текстового (Python/C++).
10. Принцип наочності: використання мультимедіа, динамічних моделей алгоритмів та скринкастів.
11. Принцип доступності: дидактична обробка складних понять (ООП, рекурсія, бази даних) для учнів середньої школи.
12. Принцип свідомості й активності: способи залучення учнів до активного кодування та розв'язання логічних задач.
13. Принцип міцності знань та особливості тренувальних вправ у програмуванні.
14. Поняття змісту освіти та чинники його формування в галузі «Інформатика».
15. Аналіз Державного стандарту базової та повної середньої освіти щодо інформатичної галузі.
16. Класифікація методів навчання за джерелом знань та їх застосування в комп'ютерному класі.
17. Пояснювально-ілюстративний та репродуктивний методи: межі використання при вивченні софту.
18. Проблемне навчання: створення проблемних ситуацій через «баги» у коді та пошук їх розв'язання.

19. Евристичний та дослідницький методи у проєктній діяльності учнів.
20. Метод проєктів як провідна технологія реалізації програми «Інформатика та програмування».
21. Урок як основна форма організації навчання: вимоги, структура, типологія.
22. Специфіка уроку-практикуму та комбінованого уроку в кабінеті інформатики.
23. Дидактичні вимоги до використання засобів ІКТ та хмарних сервісів (Google Classroom, GitHub).
24. Технологія змішаного навчання (Blended Learning): моделі «Перевернутий клас» та «Ротація станцій».
25. Позакласна робота з інформатики: хакатони, олімпіади з програмування, гуртки робототехніки.
26. Дистанційне навчання: дидактичні виклики та інструменти взаємодії.
27. Контроль у навчанні: види (попередній, поточний, тематичний, підсумковий) та їх специфіка в ІТ.
28. Методи та цифрові інструменти перевірки результатів навчання (автоматизоване тестування, Code Review).
29. Критерії оцінювання навчальних досягнень учнів з програмування (рівні: початковий, середній, достатній, високий).
30. Формувальне оцінювання: використання рефлексії та чек-листів для самоконтролю учнів.

8. ОЦІНЮВАННЯ ЗДОБУВАЧІВ ВИЩОЇ ТА ФАХОВОЇ ПЕРДВИЩОЇ ОСВІТИ ЗА НАВЧАЛЬНУ ДИСЦИПЛІНУ У МІЖНАРОДНОМУ ГУМАНІТАРНОМУ УНІВЕРСИТЕТІ

(поточний контроль, підсумковий контроль, загальний результат оцінювання)

ЕКЗАМЕН

Поточний контроль

Види роботи	Планові терміни виконання	Форми контролю та звітності	Максимальний відсоток оцінювання
Систематичність і активність роботи на практичних (лабораторних) заняттях			
1.1. Підготовка до практичних (лабораторних) занять	Відповідно до робочої програми та розкладу занять	Перевірка обсягу та якості засвоєного матеріалу під час практичних (лабораторних) занять	30
Виконання завдань для самостійного опрацювання			
1.2. Індивідуальне тестування, завдання, підготовка реферату (есе) за заданою тематикою	Відповідно до робочої програми та розкладу занять	Розгляд відповідного матеріалу під час аудиторних занять або ІКР ¹ , перевірка конспектів навчальних текстів тощо	10
Виконання індивідуальних завдань, дослідна робота здобувача			

¹ Індивідуально-консультативна робота викладача зі здобувачами

1.3. Інші види індивідуальних завдань, в т.ч. підготовка наукових публікацій, участь у роботі круглих столів, конференцій тощо.	Відповідно до робочої програми та розкладу занять	Обговорення результатів проведеної роботи під час аудиторних занять, наукових конференцій та круглих столів.	10
Разом балів за поточний контроль			50
Підсумковий контроль екзамен			50
Загальний результат оцінювання			100

Поточний контроль знань здобувачів освіти при формі контролю **екзамен** (максимум 50 балів) оцінюється за шкалою («2», «3», «4», «5»), з обов'язковим переведенням балів за кожний вид роботи таким чином:

- за підготовку до аудиторних занять (максимум 30 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості практичних (лабораторних) занять, для переведу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 6;
- за виконання завдань для самостійного опрацювання (тестування, завдання, підготовка реферату (есе) за заданою тематикою) (максимум 10 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості програмного матеріалу, що виноситься на самостійне вивчення, для переведу зазначених балів середньоарифметична оцінка множиться на коефіцієнт 2;
- за інші види індивідуальних завдань, в т.ч. підготовку наукових публікацій, участь у роботі круглих столів, конференцій тощо (максимум 10-балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості інших видів індивідуальних завдань, для переведу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 2.

КРИТЕРІЇ ОЦІНЮВАННЯ ЗА ПОТОЧНИЙ КОНТРОЛЬ

- «2»- виставляється за неправильну відповідь (письмову роботу), яка не відповідає змісту теми та свідчить про нерозуміння її основних положень;
- «3» -виставляється за відповідь (письмову роботу), яка відповідає змісту теми, але є неповною і неточною у визначенні понять, свідчить про неповне розуміння основних положень навчального матеріалу, свідчить про те, що знання студента мають фрагментарний, поверховий характер;
- «4» - оцінюється правильна і обґрунтована відповідь (письмова робота), з якої видно, що студент знає й розуміє теоретичний матеріал в обсязі навчальної теми, володіє необхідними навичками, не допускає суттєвих помилок.
- «5» - оцінюється повна, правильна, ґрунтовна відповідь (письмова робота) на запитання теми, що передбачає логічність і послідовність викладу матеріалу, володіння термінологією, показує вміння повно й глибоко використовувати теоретичні знання в практичній площині.

КРИТЕРІЇ ОЦІНЮВАННЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ ЗДОБУВАЧІВ ЗА ЕКЗАМЕН (максимум 50 балів)

Рівень знань оцінюється:

- від 40 до 50 балів - студент виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки;
- 30 до 40 балів - студент володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою

викладача швидко орієнтується і знаходить правильні відповіді;

- 20 - 30 балів - студент відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки.
- 10 - 20 балів - володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих;
- 0 - 10 балів - володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки.

КРИТЕРІЇ ОЦІНЮВАННЯ ЗА ПОТОЧНИЙ КОНТРОЛЬ

- «2»- виставляється за неправильну відповідь (письмову роботу), яка не відповідає змісту теми та свідчить про нерозуміння її основних положень;
- «3» -виставляється за відповідь (письмову роботу), яка відповідає змісту теми, але є неповною і неточною у визначенні понять, свідчить про неповне розуміння основних положень навчального матеріалу, свідчить про те, що знання студента мають фрагментарний, поверховий характер;
- «4» - оцінюється правильна і обґрунтована відповідь (письмова робота), з якої видно, що студент знає й розуміє теоретичний матеріал в обсязі навчальної теми, володіє необхідними навичками, не допускає суттєвих помилок.
- «5» - оцінюється повна, правильна, ґрунтовна відповідь (письмова робота) на запитання теми, що передбачає логічність і послідовність викладу матеріалу, володіння термінологією, показує вміння повно й глибоко використовувати теоретичні знання в практичній площині.

9. КРИТЕРІЇ ЗАГАЛЬНИХ РЕЗУЛЬТАТІВ ОЦІНЮВАННЯ ЗНАНЬ ЗДОБУВАЧІВ

Рівень знань оцінюється:

- «відмінно» / «зараховано» А - від 90 до 100 балів. Студент виявляє особливі творчі здібності, вмє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки. Був присутній на лекціях та семінарських заняттях, під час яких давав вичерпні, обґрунтовані, теоретично і практично правильні відповіді, має конспект з виконаними завданнями до самостійної роботи, презентував реферат (есе) за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;
- «добре» / «зараховано» В - від 82 до 89 балів. Студент володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні відповіді. Був присутній на лекціях та семінарських заняттях, має конспект з виконаними завданнями до самостійної роботи, презентував реферат (есе) за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;
- «добре» / «зараховано» С - від 74 до 81 балів. Студент відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи, реферату та активність у науково-дослідній роботі;
- «задовільно» / «зараховано» D - від 64 до 73 балів. Студент був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи, рефератів (есе);
- «задовільно» / «зараховано» E - від 60 до 63 балів. Студент був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки, має неповний конспект з завданнями до самостійної роботи.

- «незадовільно з можливістю повторного складання» / «не зараховано» FX – від 35 до 59 балів. Студент володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу.
- «незадовільно з обов'язковим повторним вивченням дисципліни» / «не зараховано» F – від 0 до 34 балів. Студент не володіє навчальним матеріалом.

ТАБЛИЦЯ ВІДПОВІДНОСТІ ЗАГАЛЬНИХ РЕЗУЛЬТАТІВ ОЦІНЮВАННЯ ЗНАНЬ ЗА РІЗНИМИ ШКАЛАМИ

100-бальною шкалою	Шкала за ECTS	За національною шкалою	
		екзамен	залік
90-100	A	Відмінно	Зараховано
82-89	B	Добре	
74-81	C		
64-73	D	Задовільно	
60-63	E		
35-59	Fx	Незадовільно	Не зараховано
1-34	F		

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Волкова Н. П. Педагогіка : навч. посіб. Київ : Академія, 2012. 616 с.
2. Дичківська І. М. Інноваційні педагогічні технології : навч. посіб. Київ : Академвидав, 2015. 304 с.
3. Морзе Н. В. Методика навчання інформатики : посібник : у 3 ч. Київ : Навчальна книга, 2003–2004.
4. Савченко О. Я. Дидактика початкової освіти : підручник. Київ : Грамота, 2018. 448 с.
5. Спірін О. М. Теоретичні та методичні засади навчання програмування майбутніх учителів інформатики. — Житомир: Вид-во ЖДУ ім. І. Франка, 2007.
6. Триус Ю. В. Комп'ютерно-орієнтовані методичні системи навчання математичних та інформатичних дисциплін. — Черкаси: Брама-Україна, 2005.
7. Фіцула М. М. Педагогіка : навч. посіб. Київ : Академвидав, 2022. 456 с.
8. Єрмакова С.С. Іванова О. С. Конспект лекцій з дисципліни «Основи дидактики» для здобувачів вищої освіти першого (бакалаврського) рівня освітньо-професійної програми «Інформатика та програмування», спеціальності А4.09 Середня освіта (Інформатика) денної та заочної форми навчання / С. С. Єрмакова, Іванова О. С.; [Електронне видання]. – О.: МГУ, 2025. – 94 с.
9. Єрмакова С.С. Іванова О. С. Навчально-методичні рекомендації до практичних занять та самостійної роботи з навчальної дисципліни «Основи дидактики» для здобувачів вищої освіти першого (бакалаврського) рівня освітньо-професійної програми «Інформатика та програмування»,

спеціальності А4.09 Середня освіта (Інформатика) денної та заочної форми навчання / С. С. Єрмакова, Іванова О. С.; [Електронне видання]. Одеса: МГУ, 2025. 45 с.

10. Єрмакова С.С. Іванова О. С. Конспект лекцій з дисципліни «Загальна педагогіка» для здобувачів вищої освіти першого (бакалаврського) рівня освітньо-професійної програми «Інформатика та програмування», спеціальності А4.09 Середня освіта (Інформатика) денної та заочної форми навчання / С. С. Єрмакова, Іванова О. С.; [Електронне видання]. – О.: МГУ, 2025. – 104 с.

11. Єрмакова С.С. Іванова О. С. Навчально-методичні рекомендації до практичних занять та самостійної роботи з навчальної дисципліни «Загальна педагогіка» для здобувачів вищої освіти першого (бакалаврського) рівня освітньо-професійної програми «Інформатика та програмування», спеціальності А4.09 Середня освіта (Інформатика) денної та заочної форми навчання / С. С. Єрмакова, Іванова О. С.; [Електронне видання]. Одеса: МГУ, 2025. 41 с.

Допоміжна

1. Биков В. Ю. Моделі організаційних систем відкритої освіти: Монографія. Київ: Атіка, 2008.
2. Гризун Л. Е. Дидактичні основи проектування сучасних комп'ютерних навчальних програм. Харків: ХНПУ, 2014.
3. Дичківська І. М. Інноваційні педагогічні технології. — К.: Академвидав, 2015.
4. Морзе Н. В., Барна О. В., Вембер В. П. Формувальне оцінювання: від теорії до практики. Київ: Морзе, 2021.
5. Семеріков С. О. Фундаменталізація навчання інформатичних дисциплін у вищій школі: Монографія. Кривий Ріг: Мінерал, 2009.

Нормативно-правові акти:

1. Закон України Про повну загальну середню освіту. (Зі змінами, внесеними згідно з Законом № 4265-IX від 15.06.2025). Офіційний портал Верховної Ради України. Режим доступу: <https://zakon.rada.gov.ua/laws/show/463-20#Text>.
2. Державний стандарт базової середньої освіти (Постанова КМУ № 898 від 30.09.2020 року). Офіційний сайт Міністерства освіти і науки України. Режим доступу: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/nova-ukrainska-shkola-2/derzhavniy-standart-bazovoi-serednoi-osviti>.
3. Типові освітні програми. Офіційний сайт Міністерства освіти і науки України. Режим доступу: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programi/tipovi-osvitni-programi-2>.
4. Навчальні програми на основі модельних. Офіційний сайт Міністерства освіти і науки України. Режим доступу: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programi/navchalni-prohramy-na-osnovi-modelnykh>.
5. Інформатика в школі : наук.-метод. журн. Харків : Видавнича група «Основа». URL: journal.osnova.com.ua
6. Інформаційні технології і засоби навчання : електрон. наук. фахове вид. / Ін-т цифровізації освіти НАПН України. URL: <https://journal.iitta.gov.ua>
7. Нова українська школа : сайт. URL: <https://nus.org.ua>
8. Coursera : освітня онлайн-платформа. URL: <https://www.coursera.org>
9. edX : освітня онлайн-платформа. URL: <https://www.edx.org>

Інформаційні ресурси

10. Бібліотека Міжнародного гуманітарного університету: офіційний сайт. Режим доступу: https://mgu.edu.ua/science_library
11. Бібліотека Університету Ушинського : офіційний сайт. Режим доступу: <https://library.pdpu.edu.ua>
12. Міністерство освіти і науки України: офіційний сайт. Режим доступу: <http://www.mon.gov.ua>

13. Національна бібліотека України імені В.І.Вернадського: офіційний сайт. URL: <http://www.nbuv.gov.ua/>
14. Одеська національна наукова бібліотека: офіційний сайт. Режим доступу: <http://odnb.odessa.ua/>