

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра інформаційної та кібернетичної безпеки
імені професора Володимира Бурячка

«ЗАТВЕРДЖУЮ»

Проректор з науково-педагогічної
та навчальної роботи


Олексій ЖИЛЬЦОВ
« ___ » _____ 2024 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ПРОГРАМУВАННЯ: МОВА ПРОГРАМУВАННЯ PYTHON»

для студентів

спеціальності

123 Комп'ютерна інженерія

освітнього рівня

першого (бакалаврського)

освітньої програми

123.00.01 Комп'ютерна інженерія

2023 – 2024 навчальний рік

КИЇВСЬКИЙ СТОЛИЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ БОРИСА ГРІНЧЕНКА
Код ЄДРПОУ 45307965
Програма № 3344/24
Начальник відділу моніторингу якості освіти

(підпис) (прізвище, ініціали)
« ___ » _____ 20.. 24

Розробник:

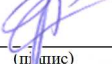
Бешко Богдан Тарасович, PhD, старший викладач кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Факультету інформаційних технологій та математики Київського столичного університету імені Бориса Грінченка.

Викладач:

Бешко Богдан Тарасович, PhD, старший викладач кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Факультету інформаційних технологій та математики Київського столичного університету імені Бориса Грінченка.


Робочу програму розглянуто і затверджено на засіданні кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка

Протокол від 03.01.2024 р. № 1

Завідувач кафедри _____  _____ Павло СКЛАДАННИЙ
(підпис)


Робочу програму погоджено з гарантом освітньої програми (керівником освітньої програми 123.00.01 Комп'ютерна інженерія)

_____.____. 2024 р.

Керівник освітньої програми _____  _____ Павло СКЛАДАННИЙ
(підпис)

Робочу програму перевірено

_____.____. 2024 р.

Заступник декана _____  _____ Євген ІВАНІЧЕНКО
(підпис)

Пролонговано:

на 20__/20__ н.р. _____ (підпис) _____ (ПІБ), «____» _____ 20__ р., протокол № ____

на 20__/20__ н.р. _____ (підпис) _____ (ПІБ), «____» _____ 20__ р., протокол № ____

на 20__/20__ н.р. _____ (підпис) _____ (ПІБ), «____» _____ 20__ р., протокол № ____

на 20__/20__ н.р. _____ (підпис) _____ (ПІБ), «____» _____ 20__ р., протокол № ____

1. Опис навчальної дисципліни

Найменування показників	Характеристика дисципліни за формами навчання	
	денна	заочна
Вид дисципліни	обов'язкова	
Мова викладання, навчання та оцінювання	українська	
Загальний обсяг кредитів / годин	3 / 90	
Курс	2	
Семестр	4	
Кількість змістових модулів з розподілом:	3	
Обсяг кредитів	3	
Обсяг годин, в тому числі:	90	
Аудиторні	42	
Модульний контроль	6	
Семестровий контроль	10	
Самостійна робота	32	
Форма семестрового контролю	іспит	

2. Мета та завдання навчальної дисципліни

Робоча навчальна програма з курсу «Програмування: Мова програмування Python» є нормативним документом Київського університету імені Бориса Грінченка, який розроблено кафедрою інформаційної та кібернетичної безпеки імені професора Володимира Бурячка на основі освітньо-професійної програми підготовки здобувачів першого (бакалаврського) рівня відповідно до навчального плану спеціальності 123 Комп'ютерна інженерія.

Робочу навчальну програму укладено згідно з вимогами Європейської кредитної трансферно-накопичувальної системи (ЄКТС) організації навчання.

Програма визначає обсяги знань, які повинен опанувати здобувач першого (бакалаврського) рівня відповідно до вимог освітньо-кваліфікаційної характеристики, алгоритму вивчення навчального матеріалу дисципліни «Програмування: Мова програмування Python» та необхідне методичне забезпечення, складові і технологію оцінювання навчальних досягнень студентів.

Навчальна дисципліна «Програмування: Мова програмування Python» складається з трьох змістових модулів: «Вступ до програмування мовою Python», «Розширене програмування мовою Python», «Прикладне використання Python». Обсяг дисципліни – 90 год (3 кредити).

Метою викладання навчальної дисципліни «Програмування: Мова програмування Python» є формування у майбутніх фахівців необхідного рівня програмування з використанням мови Python, включаючи основи синтаксису, розробку комплексних програмних рішень, використання сучасних підходів та технологій для створення веб-додатків та інтерфейсів. Програма спрямована на розвиток навичок аналізу, проектування, тестування та оптимізації програмного забезпечення, а також на ознайомлення з практичним використанням Python у реальних проектах.

Завдання:

- вивчення базових понять, розуміння синтаксису мови програмування Python, контрольних конструкцій, типів даних і структур даних, застосування метакласів, декораторів, паттернів проектування для підвищення гнучкості та масштабованості коду;
- практичне використання модулів та пакетів, організація коду за допомогою модульності, використання сторонніх бібліотек;
- вивчення основ мультипрограмування та синхронізація, введення в мультипрограмування, взаємодія з процесами та потоками, управління конкуренцією;
- оволодіння базовими навичками використання Django та Flask для створення сучасних веб-додатків;

- освоєння методів тестування та налагодження програм на основі PyTest для забезпечення якості додатків.

У результаті вивчення навчальної дисципліни формуються наступні **загальні та фахові компетентності**:

загальні компетентності:

- **ЗК 3** Здатність застосовувати знання у практичних ситуаціях;
- **ЗК 7** Вміння виявляти, ставити та вирішувати проблеми;

фахові компетентності:

- **ФК 2** Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
- **ФК 3** Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
- **ФК 7** Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.
- **ФК 11** Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.
- **ФК 12** Здатність ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів, комп'ютерних та кіберфізичних систем, мереж та їхніх компонентів шляхом використання аналітичних методів і методів моделювання.
- **ФК 13** Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій..

3. Результати навчання за дисципліною

У результаті вивчення навчальної дисципліни студент повинен

знати:

- Основи програмування та синтаксису Python.
- Структуру та основні типи даних у Python.
- Техніки роботи з колекціями, включаючи списки, кортежі, множини, та словники.
- Методи роботи з файлами, файловою системою, мережевим протоколом в Python.

уміти:

- Розробляти та впроваджувати програми на мові Python, використовуючи її стандартні бібліотеки.
- Застосовувати основні конструкції програмування, такі як умовні оператори та цикли, для рішення задач.
- Використовувати функції та модулі для структурування коду та повторного використання програмного коду.
- Аналізувати задачі та застосовувати відповідні структури даних та алгоритми для їх вирішення.
- Розробляти програми, які ефективно працюють з даними, використовуючи файлову систему та внутрішні структури даних.

та досягти наступних **програмних результатів навчання**:

- **РН 2** Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах;
- **РН 7** Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності;
- **РН 10** Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання;

4. Структура навчальної дисципліни

Тематичний план для денної форми навчання

Назва змістових модулів, тем	Усього	Розподіл годин між видами робіт					Самостійна
		Аудиторна:					
		Лекції	Семінари	Практичні	Лабораторні	Індивідуальні	
Змістовий модуль 1. Вступ до програмування мовою Python							
Тема 1. Основи програмування Python.	5	1		2			2
Тема 2. Основи синтаксису та контрольні конструкції: Введення в базові оператори, умовні конструкції та цикли.	5	1			2		2
Тема 3. Типи даних і структури даних: Огляд основних типів та структур даних.	7	1		2			4
Тема 4. Робота з ітераторами та генераторами: Вивчення ітераторів, генераторів, функцій-генераторів та виразів-генераторів.	5	1			2		2
Модульний контроль	2						
Разом	24	4		4	4		10
Змістовий модуль 2. Розширене програмування мовою Python							
Тема 5. Об'єктно-орієнтоване програмування в Python: Розгляд об'єктів, класів, класових ієрархій та абстрактних класів.	5	1		2			2
Тема 6. Метакласи. Декоратори класів. Паттерни проектування та архітектурні шаблони.	7	1			2		4
Тема 7. Виключення та модульність: Обробка виключень, використання модулів та пакетів, документування коду.	7	1		2			4
Тема 8. Мультипрограмування та синхронізація: Взаємодія з процесами, потоками, підпрограмами, реалізація паралелізму, конкуренції та синхронізації.	4	1			2		1
Модульний контроль	2						
Разом	25	4		4	4		11
Змістовий модуль 3. Прикладне використання Python							
Тема 9. Розробка графічних інтерфейсів: Основи створення GUI за допомогою бібліотеки PyQt.	5	1		2			2
Тема 10. Запуск і налаштування Django та Flask: Інсталяція фреймворку, створення проектів з використанням Django / Flask. Розробка UI з шаблонами: Створення шаблонів UI, генерація та обробка форм, валідація даних.	9	1			4		4
Тема 11. Маршрутизація та обробка запитів: Налаштування маршрутизації, CRUD операції, REST архітектура. Контролери та	7	1		4			2

Назва змістових модулів, тем	Усього	Розподіл годин між видами робіт					Самостійна
		Аудиторна:					
		Лекції	Семінари	Практичні	Лабораторні	Індивідуальні	
взаємодія з користувачем: Обробка запитів користувачів, робота з динамічними даними, куки, сесії, фільтри.							
Тема 12. Робота з моделями в Django та Flask: Використання моделей для представлення та обробки даних, міграції, встановлення відносин між моделями. Тестування і налагодження проєктів на Django / Flask: Використання фреймворку PyTest для тестування Django / Flask проєктів.	8	1			4		3
Модульний контроль	2						
Разом	31	4		6	8		11
Семестровий контроль	10						
Усього	90	12		14	16		32

5. Програма навчальної дисципліни

Змістовий модуль 1. Вступ до програмування мовою Python

Тема 1. Основи програмування мовою Python.

Історія та сфери застосування. Роль у сучасному технологічному світі. Тенденції розвитку та майбутні перспективи. Різні підходи до програмування. Інсталяція Python, використання REPL, написання та виконання скриптів. Процеси компіляції, інтерпретації та запуску коду на Python. Розбір коду на низькому рівні. Передавання аргументів у скрипт. Основні лексичні структури, базові правила синтаксису та деякі стандартні функції.

Тема 2. Основи синтаксису та контрольні конструкції: Введення в базові оператори, умовні конструкції та цикли.

Детальний аналіз синтаксичної структури мови Python. Вивчення базових операторів, що забезпечують виконання арифметичних та логічних операцій. Дослідження циклічних конструкцій. Основна увага приділяється формуванню навичок коректного застосування цих структур для оптимізації логіки програм та ефективного управління потоком виконання.

Тема 3. Типи даних і структури даних: Огляд основних типів та структур даних.

Тема включає класифікацію та властивості типів даних у Python: примітивні (цілі числа, числа з плаваючою точкою, булеві значення) та складені (рядки, списки, кортежі, словники, множини). Опис механізмів створення, індексації, зрізів, ітерації та модифікації даних у колекціях. Розгляд алгоритмічної складності операцій з різними структурами даних. Вивчення використання типів даних для реалізації алгоритмів та розв'язання задач.

Тема 4. Робота з ітераторами та генераторами: Вивчення ітераторів, генераторів, функцій-генераторів та виразів-генераторів.

Аналізується концепція ітераторів та механізми їх створення та використання у Python для ефективноної обробки колекцій даних. Вивчення генераторів, як способу легкого створення ітераторів через функції з використанням оператора yield. Огляд функцій-генераторів, які дозволяють лаконічно вирішувати задачі, пов'язані з лінивими обчисленнями. Розгляд виразів-генераторів для створення компактного коду, який здійснює ітерацію по даних з мінімальними затратами пам'яті.

Змістовий модуль 2. Розширене програмування мовою Python

Тема 5. Об'єктно-орієнтоване програмування в Python: Розгляд об'єктів, класів, класових ієрархій та абстрактних класів.

Концепції об'єктно-орієнтованого програмування (ООП) у Python. Вивчається створення класів, інкапсуляція даних та методів. Розглядаються наслідування для розширення класів, поліморфізм для використання однієї інтерфейсної форми з різними підлягаючими типами. Описуються абстрактні класи та інтерфейси, що використовуються для формування загального шаблону класу. Тема покриває розуміння класових ієрархій та використання композиції для більш гнучкої альтернативи наслідуванню.

Тема 6. Метакласи. Декоратори класів. Паттерни проектування та архітектурні шаблони.

Вивчення метакласів: аналіз механізму створення класів в Python, вплив метакласів на процес опису нових типів даних. Розгляд декораторів класів: застосування декораторів для модифікації або розширення функціональності класів перед їхнім створенням. Аналіз паттернів проектування: огляд та використання основних шаблонів проектування (Singleton, Factory, Observer та ін.) в контексті Python для рішення типових задач програмування. Вивчення архітектурних шаблонів: застосування архітектурних паттернів (MVC, MVVM) для організації коду великих проектів.

Тема 7. Виключення та модульність: Обробка виключень, використання модулів та пакетів, документування коду.

Аналіз механізмів обробки виключень для контролю виконання програм у випадках виникнення помилок. Вивчення системи модулів та пакетів Python для організації коду в масштабних проектах. Огляд методів документування коду, включно з використанням docstrings, для підвищення його читабельності та підтримки.

Тема 8. Мультипрограмування та синхронізація: Взаємодія з процесами, потоками, підпрограмами, реалізація паралелізму, конкуренції та синхронізації.

Вивчення концепцій мультипрограмування: процеси, потоки, корутини. Аналіз методів взаємодії між процесами та потоками в Python. Реалізація паралельного виконання коду за допомогою модулів `threading` та `multiprocessing`. Введення в концепції конкуренції та блокувань для уникнення гонки даних. Застосування механізмів синхронізації, таких як семафори та мьютекси, для контролю доступу до спільних ресурсів.

Змістовий модуль 3. Прикладне використання Python

Тема 9. Розробка графічних інтерфейсів: Основи створення GUI за допомогою бібліотеки PyQt.

Вивчення основ створення графічних інтерфейсів користувача (GUI) в Python за допомогою бібліотеки PyQt. Ознайомлення з класами та методами PyQt для розробки візуальних компонентів: вікна, кнопки, текстові поля, меню. Реалізація подій та сигналів для взаємодії з користувачем. Структурування додатків з використанням моделі поділу на MVC (Model-View-Controller). Проектування та налаштування віджетів для створення інтуїтивно зрозумілого та функціонального інтерфейсу.

Тема 10. Запуск і налаштування Django та Flask: Інсталяція фреймворку, створення проектів з використанням Django / Flask. Розробка UI з шаблонами: Створення шаблонів UI, генерація та обробка форм, валідація даних.

Огляд процедур інсталяції веб-фреймворків Django та Flask для розробки веб-додатків на Python. Вивчення кроків створення нових проектів, включаючи налаштування середовища розробки, структуру проекту та основні конфігурації. Опис методів запуску розробницького сервера, додавання та конфігурації нових додатків у проекті. Розгляд базових принципів роботи з моделями, відображеннями та шаблонами у Django, та маршрутизації, відображення та шаблонів у Flask. Акцент на імпортантності розуміння архітектурних відмінностей між фреймворками для ефективного вибору інструментарію відповідно до вимог проекту.

Аналіз методологій створення користувацьких інтерфейсів через шаблони в Python, з

використанням фреймворків Django та Flask. Огляд процесу дефінування шаблонів UI для ефективного представлення даних та інтерактивності з користувачем. Розгляд механізмів генерації та обробки форм, включаючи передачу даних від клієнта до сервера. Вивчення методів валідації даних на стороні сервера для забезпечення їхньої коректності та безпеки. Розбір практик оптимізації взаємодії користувача з веб-додатком через використання передових технік шаблонізації.

Тема 11. Маршрутизація та обробка запитів: Налаштування маршрутизації, CRUD операції, REST архітектура. Контролери та взаємодія з користувачем: Обробка запитів користувачів, робота з динамічними даними, куки, сесії, фільтри.

Вивчення принципів маршрутизації в рамках веб-фреймворків Django та Flask для реалізації взаємодії з користувачем через HTTP запити. Розгляд конфігурації маршрутів для обробки вхідних запитів та направлення їх до відповідних контролерів або відображень. Детальний аналіз CRUD (Create, Read, Update, Delete) операцій як фундаментальних методів взаємодії з даними в базах даних через веб-інтерфейс. Введення в архітектуру REST (Representational State Transfer) для розробки масштабованих веб-сервісів, що підтримують стандартизований доступ до ресурсів. Наголос на значенні коректного налаштування маршрутизації та обробки запитів для забезпечення безпеки, продуктивності та зручності використання веб-додатків.

Огляд розробки контролерів для ефективної обробки вхідних запитів від користувачів у веб-додатках на Python, використовуючи Django та Flask. Вивчення механізмів приймання, обробки та відповіді на запити, враховуючи параметри запиту, тіло запиту та інші HTTP-атрибути. Аналіз роботи з динамічними даними, отриманими в результаті взаємодії користувача з веб-додатком. Розгляд використання куків та сесій для зберігання стану користувача під час навігації по веб-додатку. Введення у фільтри для передобробки запитів або відповідей з метою реалізації додаткової логіки безпеки, автентифікації чи оптимізації відповідей.

Тема 12. Робота з моделями в Django та Flask: Використання моделей для представлення та обробки даних, міграції, встановлення відносин між моделями. Тестування і налагодження проектів на Django / Flask: Використання фреймворку PyTest для тестування Django / Flask проектів.

Вивчення використання моделей у Django та Flask для абстракції та взаємодії з базою даних. Аналіз оголошення моделей, включаючи визначення полів та їх типів. Розгляд механізму міграцій для внесення змін у схему бази даних. Вивчення створення відносин між моделями, таких як один-до-багатьох, багато-до-багатьох, та один-до-одного, для моделювання складних структур даних. Обговорення методів запитів до бази даних через ORM для створення, читання, оновлення та видалення записів.

Ознайомлення з принципами тестування та налагодження веб-проектів на основі Django та Flask за допомогою PyTest. Вивчення створення тестових випадків для перевірки функціональності, інтеграційного тестування компонентів та мок-тестування залежностей. Розгляд конфігурації PyTest для веб-додатків, включаючи налаштування фікстур для ініціалізації тестового середовища. Аналіз практик налагодження коду, виявлення помилок та оптимізації продуктивності. Обговорення стратегій забезпечення якості коду та підвищення надійності веб-додатків.

6. Контроль навчальних досягнень

Навчальні досягнення студентів з дисципліни оцінюються за модульно-рейтинговою системою, в основу якої покладено принцип поопераційної звітності, обов'язковості модульного контролю, накопичувальної системи оцінювання рівня знань, умінь та навичок, розширення кількості підсумкових балів до 100.

Оцінка за кожний змістовий модуль включає бали за поточну роботу студента на практичних та лабораторних заняттях, за виконання індивідуальних завдань, за модульну контрольну роботу. Виконання модульних контрольних робіт здійснюється в електронному

вигляді. Модульний контроль знань студентів здійснюється після завершення вивчення навчального матеріалу змістового модуля.

У процесі оцінювання навчальних досягнень студентів застосовуються такі методи:

- *Методи усного контролю*: індивідуальне опитування, фронтальне опитування, співбесіда, екзамен.
- *Комп'ютерного контролю*: тестові програми.
- *Методи самоконтролю*: уміння самостійно оцінювати свої знання, самоаналіз.

Кількість балів за роботу з теоретичним матеріалом, на практичних заняттях, під час виконання самостійної роботи залежить від дотримання таких вимог:

- систематичність відвідування занять;
- своєчасність виконання навчальних і індивідуальних завдань;
- повний обсяг їх виконання;
- якість виконання навчальних і індивідуальних завдань;
- самостійність виконання;
- творчий підхід у виконанні завдань;
- ініціативність у навчальній діяльності;
- виконання тестових завдань.

Контроль успішності студентів з урахуванням поточного і підсумкового оцінювання здійснюється відповідно до навчально-методичної карти дисципліни, де зазначено види контролю і кількість балів за видами. Систему рейтингових балів для різних видів контролю та порядок їх переведення у національну (4-бальну) та європейську (ECTS) шкалу подано нижче у таблицях.

Розрахунок рейтингових балів за видами поточного (модульного) контролю

Вид діяльності студента	Максимальна к-сть балів за одиницю	Модуль 1		Модуль 2		Модуль 3	
		Кількість одиниць	Максимальна кількість балів	Кількість одиниць	Максимальна кількість балів	Кількість одиниць	Максимальна кількість балів
Відвідування лекцій	1	2	2	2	2	2	2
Відвідування практичних занять	1	2	2	2	2	3	3
Відвідування лабораторних занять	1	2	2	2	2	4	4
Робота на практичному занятті	10	2	20	2	20	3	30
Лабораторна робота (в тому числі допуск, виконання, захист)	10	2	20	2	20	4	40
Виконання завдань для самостійної роботи	5	1	5	1	5	1	5
Виконання модульної роботи	25	1	25	1	25	1	25
	Разом	-	76	-	76	-	109
Максимальна кількість балів: 261							
Розрахунок коефіцієнта: $261/100=2,61$							

Завдання для самостійної роботи та критерії її оцінювання

Самостійна робота є видом поза аудиторної індивідуальної діяльності студента, результати якої використовуються у процесі вивчення програмового матеріалу навчальної дисципліни та містить результати дослідницького пошуку, відображає певний рівень його навчальної компетентності.

Перелік тем та оцінювання самостійної роботи студента

№ з/п	Назва теми	Кількість годин	Бали
	Змістовий модуль 1. Вступ до програмування мовою Python	10	5

1	Розробити консольну програму на Python, яка включає використання базових операторів, умовних конструкцій та циклів для обчислення і виведення результатів різних математичних завдань (наприклад, факторіал числа, робота з геометричними фігурами). Програма має також демонструвати роботу з різними типами даних і структурами даних (списки, кортежі, словники), а також містити приклади створення та використання ітераторів та генераторів для обходу колекцій.	10	5
Змістовий модуль 2. Розширене програмування мовою Python		11	5
2	Створити об'єктно-орієнтовану систему управління базою даних книг у бібліотеці, яка використовує класи для представлення книг, авторів та читачів. Включити в систему метакласи для логування створення об'єктів, декоратори для валідації даних та імплементуйте базові паттерни проектування для оптимізації взаємодії між об'єктами. Реалізувати обробку виключень для захисту від некоректного вводу даних і використовуйте модулі для організації коду. Додати мультипоточність для одночасного виконання декількох запитів до системи.	11	5
Змістовий модуль 3. Прикладне використання Python		11	5
3	Розробити веб-додаток для онлайн-замовлення книг із використанням Django або Flask, що включає розробку графічного інтерфейсу користувача за допомогою PyQt або веб-шаблонів. Інтегрувати систему управління базою даних книг з модулю 2 для зберігання інформації про книги, замовлення та користувачів. Впровадити маршрутизацію, CRUD операції для керування даними через веб-інтерфейс, і використовуйте REST архітектуру для розробки API. Забезпечити взаємодію з користувачами через обробку запитів, сесій, куків. Включити тестування та налагодження веб-додатку з використанням PyTest, щоб переконатися у його надійності та ефективності.	11	5
Разом		32	15

Критерії оцінювання самостійної роботи студента

№ п/п	Критерії оцінювання роботи	Максимальна кількість балів за кожним критерієм
1	Критичний аналіз суті та змісту першоджерел. Виклад фактів, ідей, результатів досліджень в логічній послідовності. Аналіз сучасного стану дослідження проблеми, розгляд тенденцій подальшого розвитку даного питання.	2 бали
2	Доказовість висновків, обґрунтованість власної позиції, пропозиції щодо розв'язання проблеми, визначення перспектив дослідження	2 бали
3	Дотримання вимог щодо технічного оформлення	1 бал
Разом		5 балів

Форми проведення модульного контролю та критерії оцінювання

Модульний контроль здійснюється відповідно до навчально-методичної карти дисципліни та перевіряє рівень досягнення результатів навчання студентів. Форма проведення – комп'ютерний тест, що складається 15 запитань закритої та відкритої форм.

Модульна контрольна робота оцінюється у 25 балів.

Форми проведення семестрового контролю та критерії оцінювання

Семестрове (підсумкове) оцінювання здійснюється у формі іспиту, умовою отримання якого є отриманням студентом 35 балів (з врахуванням коефіцієнту) за результатами поточного контролю.

Форма проведення екзамену – комп'ютерний тест. Тест містить 20 тестових питань закритого типу (вибір правильної відповіді із запропонованих варіантів), які передбачають

автоматичну (комп'ютерну) перевірку і оцінюються по 2 бали кожне. Екзамен оцінюється у 40 балів.

Орієнтовний перелік питань для семестрового контролю

1. Основні області використання Python?
2. Як компілюється та виконується програма на Python?
3. Що таке змінна в Python?
4. Які типи даних є в Python?
5. Як створити і використати список в Python?
6. Що таке цикл у Python і які типи циклів існують?
7. Які умовні оператори доступні в Python?
8. Як працювати з словниками в Python?
9. Що таке ітератор і як він використовується в Python?
10. Як створити генератор в Python і для чого він потрібен?
11. Що таке клас у Python і як створити його екземпляр?
12. Як реалізувати наслідування класів в Python?
13. Що таке метакласи і як вони застосовуються?
14. Як використовуються декоратори класів у Python?
15. Які паттерни проектування найчастіше використовуються в Python?
16. Як обробляти виключення в Python?
17. Як організувати код у модулі та пакети?
18. Як документувати код в Python?
19. Як використовувати потоки і процеси для мультипрограмування?
20. Що таке синхронізація в контексті мультипрограмування?
21. Як створити GUI за допомогою бібліотеки PyQt?
22. Як інсталиувати та налаштувати Django або Flask для розробки веб-додатків?
23. Як створити і валідувати форми в Django або Flask?
24. Як налаштувати маршрутизацію в Django або Flask?
25. Як обробляти користувацькі запити та управляти сесіями в Django або Flask?
26. Як тестувати Django або Flask проекти з використанням PyTest?

Шкала відповідності оцінок

Рейтингова оцінка	Сума балів за всі види навчальної діяльності	Значення оцінки
A	90-100	Відмінно — відмінний рівень знань (умінь) в межах обов'язкового матеріалу з, можливими, незначними недоліками
B	82-89	Дуже добре - достатньо високий рівень знань (умінь) в межах обов'язкового матеріалу без суттєвих (грубих) помилок
C	75-81	Добре - в цілому добрий рівень знань (умінь) з незначною кількістю помилок
D	69-74	Задовільно - посередній рівень знань (умінь) із значною кількістю недоліків, достатній для подальшого навчання або професійної діяльності
E	60-68	Достатньо - мінімально можливий допустимий рівень знань (умінь)
FX	35-59	Незадовільно з можливістю повторного складання - незадовільний рівень знань, з можливістю повторного перескладання за умови належного самостійного доопрацювання
F	1-34	Незадовільно з обов'язковим повторним вивченням курсу - досить низький рівень знань (умінь), що вимагає повторного вивчення дисципліни

7. Навчально-методична картка дисципліни

Разом: 90 год., лекції – 12 год., практичні заняття – 14 год., лабораторні роботи – 16 год., модульний контроль – 6 год., самостійна робота – 17 год.

Модулі (назви, бали)	Змістовий модуль 1. Вступ до програмування мовою Python (76 балів)		Змістовий модуль 2. Розширене програмування мовою Python (76 балів)		Змістовий модуль 3. Прикладне використання Python (109 балів)	
Лекції (теми, бали)	№ 1. Основи програмування мовою Python. Основи синтаксису та контролюні конструкції: Введення в базові оператори, умовні конструкції та цикли (1 бал)		№ 2. Типи даних і структури даних: Огляд основних типів та структур даних. Робота з ітераторами та генераторами: Вивчення ітераторів, генераторів, функцій-генераторів та виразів-генераторів (1 бал)		№ 5. Розробка графічних інтерфейсів: Основи створення GUI за допомогою бібліотеки PyQt. Запуск і налаштування Django та Flask: Інсталяція фреймворку, створення проєктів з використанням Django / Flask. Розробка UI з шаблонами: Створення шаблонів UI, генерація та обробка форм, валідація даних (1 бал)	
Практичні, заняття (теми, бали)	№ 1 Розробка програмних модулів з використанням базових операторів, умовних конструкцій та циклів в Python C# (11 балів)		№ 3 Застосування метакласів, декораторів та паттернів проектування для розширення можливостей класів у Python (11 балів)		№ 6. Маршрутизація та обробка запитів: Налаштування маршрутизації, CRUD операції, REST архітектура. Контролери та взаємодія з користувачем: Обробка запитів користувачів, робота з динамічними даними, куки, сесії, фільтри. Робота з моделями в Django та Flask: Використання моделей для представлення та обробки даних, міграції, встановлення відносин між моделями. Тестування і налагодження проєктів на Django / Flask: Використання фреймворку PyTest для тестування Django / Flask проєктів (1 бал)	
Лабораторні заняття (теми, бали)	№ 1 Імплементация та маніпуляції основними типами та структурами даних у Python (11 балів)		№ 4 Розробка мультиточкових додатків з використанням синхронізації для управління конкуренцією в Python (11 балів)		№ 5 Ініціалізація та конфігурація веб-проєктів з Django та Flask (11 балів)	
Самостійна робота	Самостійна робота (5 балів)		Самостійна робота (5 балів)		Самостійна робота (5 балів)	
Поточний контроль	Модульна контрольна робота 1 (25 балів)		Модульна контрольна робота 2 (25 балів)		Модульна контрольна робота 3 (25 балів)	
Підсумковий контроль (вид, бали)	Іспит (40 балів)					

8. Рекомендовані джерела

Основна (базова):

1. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник для студ. спеціальності 122 "Комп'ютерні науки", спеціалізації "Інформаційні технології в біології та медицині" / А. В. Яковенко ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,59 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 195 с.
2. Кренивч А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник – К.: ВПЦ "Київський Університет", 2020. – 152 с.
3. Юрченко І.В., Сікора В.С. Програмування мовою Python. навчальний посібник – Чернівці. Чернівецький національний університет 2022. – 104 с.

Додаткова:

4. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
5. Висоцька В.А., Оборська О.В. Python: алгоритмізація та програмування: навчальний посібник – Львів: Видавництво «Новий Світ – 2000», 2024. – 514 с. ISBN 978-617-7519-74-3
6. The Python Coding Book: <https://thepythoncodingbook.com/>
7. Посібник Python. URL: <https://docs.python.org/uk/3/tutorial/index.html>