

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Декан факультету комп'ютерних
інформаційних технологій
Ігор ЯКИМЕНКО
« 08 » 20 24 р.



ЗАТВЕРДЖУЮ
Проректор з науково-педагогічної
роботи
Віктор ОСТРОВЕРХОВ
« 08 » 20 24 р.



РОБОЧА ПРОГРАМА
з дисципліни
«ПРОГРАМУВАННЯ II»

Ступінь вищої освіти – **бакалавр**
Галузь знань – **12 Інформаційні технології**
Спеціальність – **125 Кібербезпека та захист інформації**
Освітньо-професійна програма – **Кібербезпека**

кафедра кібербезпеки

Форма навчання	Курс	Семестр	Лекції (год.)	Практ. (год.)	ІРС (год.)	Тренінг (год.)	СРС (год.)	Разом (год.)	Іспит (сем)
Денна	2	3	46	60	6	14	114	240	3

30.08.2024

Робоча програма складена на основі освітньо-професійної програми підготовки бакалавра галузі знань 12 Інформаційні технології спеціальності – 125 Кібербезпека та захист інформації, затвердженої на засіданні Вченою радою ЗУНУ, протокол № 10 від 23.06.2023 р.

Робочу програму склав доцент кафедри кібербезпеки, к.т.н., доцент Івасьєв Степан Володимирович.

Робоча програма затверджена на засіданні кафедри кібербезпеки, протокол № 1 від 26.08.2024 р.

Завідувач кафедри
кібербезпеки



Василь ЯЦКІВ

Розглянуто та схвалено групою забезпечення спеціальності кібербезпека та захист інформації, протокол № 1 від 30.08.2024 р.

Керівник групи
забезпечення спеціальності



Василь ЯЦКІВ

Гарант ОП



Михайло КАСЯНЧУК

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1. Опис дисципліни «Програмування II»

Дисципліна «Програмування II»	Галузь знань, спеціальність, СВО	Характеристика навчальної дисципліни
Кількість кредитів – 5	Галузь знань - 12 - «Інформаційні технології»	Статус дисципліни - обов'язкова Мова навчання - українська
Кількість залікових модулів – 5	Спеціальність – 125 - «Кібербезпека та захист інформації»	Рік підготовки: 2 Семестр: 3
Кількість змістових модулів – 2	Ступінь вищої освіти – бакалавр	Лекції: 46 год. Лабораторні заняття: 60 год.
Загальна кількість годин – 240		Самостійна робота: 114 год. Тренінг – 6 год. Індивідуальна робота: 14 год.
Тижневих годин – 16год. з них аудиторних – 7год.		Вид підсумкового контролю – екзамен.

2. Мета і завдання вивчення дисципліни «Програмування II»

2.1. Мета вивчення дисципліни.

Метою викладання дисципліни «Програмування II» є навчити студентів розробляти програми мовою Python з використанням структурно-модульного методу програмування.

2.2 Завдання вивчення дисципліни:

Основне завдання курсу дати студентам теоретичну та практичну підготовку з основ функціонального та об'єктно-орієнтованого програмування на Python.

2.3. Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни:

Здатність до використання інформаційно-комунікаційних технологій, сучасних методів і моделей інформаційної безпеки та/або кібербезпеки.

Здатність до використання програмних та програмно-апаратних комплексів засобів захисту інформації в інформаційно-телекомунікаційних (автоматизованих) системах.

2.4. Передумови для вивчення дисципліни.

Вивчення курсу «Програмування II» передбачає наявність систематичних та ґрунтовних знань із суміжних курсів «Основи програмування», «Вища математика», «Архітектура комп'ютерів і систем» та ін.

2.5. Результати навчання

Використовувати сучасне програмно апаратне забезпечення інформаційно комунікаційних технологій. Використовувати програмні та програмно апаратні комплекси захисту інформаційних ресурсів. Використовувати сучасне програмно апаратне забезпечення інформаційно комунікаційних технологій.

2.6. Завдання лекційних занять

Мета проведення лекцій полягає у викладанні основних теоретичних положень, синтаксису та алгоритмічних конструкцій мовою програмування python. На лекційних заняттях будуть розглянуті найпоширеніші практики для опрацювання складних структур даних та побудови багатомодульного програмного засобу.

2.7. Завдання проведення практичних занять

Проведення практичних занять, як одна з основних форм навчального процесу, передбачає поглиблення розуміння і застосування на практиці основних базових конструкцій та структур даних, методів побудови програмних засобів та реалізації повноцінного програмного засобу.

3. Програма навчальної дисципліни «Програмування II»

Змістовий модуль 1. Основи мови Python.

Тема 1. Типи об'єктів мови Python.

Числа. Базові числові типи. Рядки. Літерали рядків. Рядки в дії. Строкові методи. Вирази

форматування рядків. Метод форматування рядків. Загальні категорії типів Списки. Словники. Кортежі. Файли. Ієрархії типів даних в мові Python. Інші базові типи.

Література: 1, 7, 12.

Тема 2. Введення в інструкції мови Python.

Структура програми на мові Python. Інструкції присвоювання. Інструкції виразів. Операція print.

Література: 1, 2, 6.

Тема 3. Умовна інструкція if і синтаксичні правила.

Умовні інструкції if. Синтаксичні правила мови Python. Перевірка істинності. Тримісний вираз if / else.

Література: 2, 5, 11.

Тема 4. Цикли while і for.

Цикли while, break, continue, pass і else. Цикл for. Правила програмування циклів.

Література: 1, 5, 7.

Тема 5. Ітерації і генератори.

Ітератори. Генератори списків. Нові об'єкти в Python 3.0 .

Література: 2, 4, 12.

Тема 6. Основи функцій.

Створення функцій. Визначення та виклики функцій.

Література: 1, 2, 5.

Тема 7. Області видимості.

Області видимості в мові Python. Інструкція global. Області видимості і вкладені функції. Інструкція nonlocal.

Література: 1, 3, 6, 8.

Тема 8. Аргументи.

Передача аргументів. Спеціальні режими зіставлення аргументів. Функція пошуку мінімуму. Універсальні функції для роботи з множинами. Імітація функції print в Python 3.0

Література: 1, 2, 6, 12.

Змістовий модуль 2. Модулі в мові в Python.

Тема 9. Розширені можливості функцій.

Концепції проектування функцій. Рекурсивні функції. Функції - це об'єкти: атрибути і анотації. Анонімні функції: lambda. Відображення функцій на послідовності: map. Засоби функціонального програмування: filter і reduce.

Література: 1, 5, 9.

Тема 10. Ітератори і генератори. Генератори списків: функціональні інструменти. Синтаксис генераторів в 3.0. Хронометраж ітераційних альтернатив. Типові помилки при роботі з функціями.

Література: 1, 5, 10.

Тема 11. Основи програмування модулів.

Навіщо потрібні модулі? Як працює імпорт. Шлях пошуку модулів. Створення модуля. Використання модулів. Простори назв модулів. Перезавантаження модулів.

Література: 1, 4, 5, 11.

Тема 12. Пакети модулів.

Основи імпортування пакетів. Приклад імпортування пакета. Коли використовується операція імпортування пакетів? Імпорт пакету.

Література: 2, 4, 5.

Тема 13. Додаткові можливості модулів.

Приховування даних в модулях. Змішані режими використання: `__name__` і `__main__`. Зміна шляху пошуку модулів. Розширення `as` для інструкцій `import` і `from`. Імпорт модулів по імені у вигляді рядка. Перезавантаження модулів. Концепції проектування модулів. Типові проблеми при роботі з модулями.

Література: 4, 3, 8.

Тема 14. Основи об'єктно-орієнтованого програмування. Навіщо потрібні класи? Класи генерують безліч екземплярів об'єктів. Класи адаптуються за допомогою успадкування. Класи можуть перевизначати оператори мови Python.

Література: 3, 6, 7.

Тема 15. Програмування класів.

Інструкція class. Методи. Спадкування. Простори назв. Класи і модулі.

Література: 4-6, 10.

4. Структура залікового кредиту з дисципліни «Програмування II»

	Кількість годин						Контрольні заходи
	Лекції	Лаборат. заняття	ІР С	СРС	Тренінг		
<i>Змістовий модуль 1. Основи мови Python</i>							
Тема 1. Типи об'єктів мови Python.	2	4		5	2	Поточне опитування	
Тема 2. Введення в інструкції мови Python	2	4		5			
Тема 3. Умовна інструкція if і синтаксичні правила.	2	4		5	2		
Тема 4. Цикли while і for.	2	4		5			
Тема 5. Ітерації і генератори.	2	4	1	6	2		
Тема 6. Основи функцій.	4	4	1	6			
Тема 7. Області видимості.	4	4	1	6	2		
Тема 8. Аргументи.	4	4		6			
<i>Змістовий модуль 2. Модулі в мові Python</i>							
Тема 9. Розширені можливості функцій	4	4		10	2	Поточне опитування	
Тема 10. Ітератори і генератори.	4	4		10			
Тема 11. Основи програмування модулів.	4	4	1	10			
Тема 12. Пакети модулів.	4	4	1	10	2		
Тема 13. Додаткові можливості модулів.	2	4	1	10			
Тема 14. Основи об'єктно-орієнтованого програмування.	2	4		10	2		
Тема 15. Програмування класів.	4	4		10			
Разом	46	60	6	114	14	Іспит	

5. Тематика лабораторних робіт

Лабораторна робота № 1.

Тема: Змінні, значення і їх типи. Присвоєння значення. Введення значень з клавіатури.

Мета: Отримати практичні навички роботи зі змінними, значеннями та їх типами.

Питання для обговорення: 1. Змінні, значення і їх типи. 2. Присвоєння значення. 3. Введення значень з клавіатури.

Література: 1, 2, 10.

Лабораторна робота № 2.

Тема: Варіанти виводу тексту на екран.

Мета: Отримати практичні навички роботи з операторами введення-виведення.

Питання для обговорення: 1. Форматований ввід/вивід. 2. Псевдографіка. 3. Робота зі стрічками. 4. Парсинг тексту.

Література: 2, 4, 8.

Лабораторна робота № 3.

Тема: Вбудовані операції і функції. Основні алгоритмічні конструкції.

Мета: Отримати практичні навички роботи із функціями та вбудованими операціями.

Питання для обговорення: 1. Зарезервовані слова. 2. Алгоритмічні конструкції. 3. Побудова

алгоритмів.

Література: 1, 5, 7.

Лабораторна робота № 4.

Тема: Оператори умов. Умови рівності / нерівності.

Мета: Отримати практичні навички роботи із конструкціями переходу.

Питання для обговорення: 1. Логічні вирази. Операції над логічними виразами. 2. Конструкції переходу. 3. Безумовний перехід.

Література: 1, 6, 8.

Лабораторна робота № 5.

Тема: Цикли і лічильники. Використання команд break, continue.

Мета: Отримати практичні навички роботи із лічильниками, спеціальними операторами циклу.

Питання для обговорення: 1. Поняття лічильника. 2. Види циклів. 3. Вихід з циклу. 4. Вічні цикли.

Література: 2, 3, 12.

Лабораторна робота № 6.

Тема: Визначення функцій. Установки та аргументи. Виклики функцій. Оператор повернення return.

Мета: Отримати практичні навички роботи із користувацькими функціями.

Питання для обговорення: 1. Аргументи функції. 2. Область видимості функції. 3. Оператор повернення.

Література: 2, 4, 9.

Лабораторна робота № 7.

Тема: Основні стандартні модулі і пакети в Python.

Мета: Отримати практичні навички роботи із модулями та пакетами в Python.

Питання для обговорення: 1. Основні модулі. 2. Пакети в Python. 3. Підключення пакетів.

Література: 2, 4, 8.

Лабораторна робота № 8.

Тема: Імпорт модулів. Створення власних модулів і їх імпортування. Спеціалізовані модулі та додатки.

Мета: Отримати практичні навички створення модулів та їхній імпорт.

Питання для обговорення: 1. Можливості створення власних модулів. 2. Можливості імпорту модулів. 3. Набір спеціалізованих модулів.

Література: 1, 2, 10.

Лабораторна робота № 9.

Тема: Різні типи послідовностей і загальні для них оператори.

Мета: Отримати практичні навички створення модулів та їхній імпорт.

Питання для обговорення: 1. Різні види послідовностей. 2. Оператори для роботи з послідовностями. 3. Оператори порівняння послідовностей. 4. Оператори друку.

Література: 5, 6, 8, 11.

Лабораторна робота № 10.

Тема: Робота зі списками. Вкладені списки. Спеціальні оператори, функції для них.

Мета: Отримати практичні навички роботи зі списками.

Питання для обговорення: 1. Списки. 2. Робота зі списками. 3. Вкладені списки. 4. Спеціальні оператори.

Література: 1, 3, 9.

Лабораторна робота № 11.

Тема: Робота зі словниками. Методи словників. Матриці.

Мета: Отримати практичні навички роботи зі словником та матрицями.

Питання для обговорення: 1. Створення словників. 2. Опрацювання словників.

3. Словники, як аргументи.
Література: 1, 4, 10.

Лабораторна робота № 12.

Тема 11. Мережні додатки на Python. Робота із сокетами. Модулі для клієнта WWW.

Мета: Отримати практичні навички роботи із мережевими можливостями мови.

Питання для обговорення: 1. Можливості мережевої взаємодії. 2. Побудова клієнта. 3. Побудова сервера.

Література: 5, 6, 12.

6. Самостійна робота

Самостійна робота студента передбачає виконання практичного індивідуального завдання, яке виконується кожним студентом. Тематика самостійної роботи охоплює основні теми курсу. Орієнтовні варіанти завдань:

1. Основні стандартні модулі Python. Поняття модуля. Модулі в Python.
2. Стандартні бібліотеки Python.
3. Елементи функціонального програмування. Що таке функціональне програмування? Функціональна програма.
4. Функція: визначення і виклик. Рекурсія.
5. Чисельні алгоритми. Матричні обчислення. Модуль Numeric.
6. Створення масиву. Зрізи. Універсальні функції. Функції модуля Numeric.
7. Функції для роботи з масивами.
8. Обробка текстів. Рядки. Кодування Python-програми.
9. Рядкові літерали. Операції над рядками. Модуль string. Методи рядків.
10. Регулярні вирази. Синтаксис регулярного виразу.
11. Методи об'єкта-шаблону. Приклади шаблонів. Налаштування регулярних виразів.

Приклади застосування регулярного виразу.

12. Об'єктно-орієнтоване програмування. Основні поняття.
13. Абстракція і декомпозиція. Об'єкти. Типи і класи. Інкапсуляція.
14. Відносини між класами. Спадкування.
15. Статичний метод. Метод класу. Метакласи. Мультиметоди.

7. Тренінг з дисциплін.

Порядок проведення тренінгу:

1. Вступна частина проводиться з метою ознайомлення студентів з темою тренінгу.
2. Організаційна частина полягає у створенні робочого настрою у колективі студентів.
3. Практична частина реалізується шляхом виконання завдань з певних проблемних питань теми тренінгу.
4. Підведення підсумків. Обговорення результатів виконаних завдань. Обмін думками з питань, що виносились на тренінг.

Тематика тренінгу: Використання спеціалізованих бібліотек та побудова багатомодульних програмних засобів.

Мета тренінгу: забезпечення студентів теоретичними знаннями та практичними навичками використання спеціалізованих бібліотек та побудови багатомодульних програмних засобів.

Завдання тренінгу: презентація результатів, за орієнтовною тематикою:

1. Створення словника використовуючи XML.
2. Приклади роботи з послідовностями.
3. Реалізація перевантаження функцій.
4. Управління контекстом виконання.
5. Генератор чисел Фібоначчі.
6. Альтернативний синтаксис доступу до елементів словника.
7. Функтор з генерацією байткода.
8. Транспонування матриці.
9. Знаходження Факторіал.
10. Рішення квадратного рівняння.
11. Шифрування Цезаря.

12. Шифрування Віженера.
13. Шифрування заміною.
14. Хешування файлів.

8. Методи навчання.

У навчальному процесі застосовуються: лекції, в тому числі з використання мультимедійного проєктора та інших ТЗН; практичні роботи, індивідуальні заняття; робота в Інтернет.

9. Засоби оцінювання та методи демонстрування результатів навчання

У процесі вивчення дисципліни «Програмування П» використовуються наступні методи оцінювання навчальної роботи студента:

- поточне тестування та опитування;
- підсумкове тестування за кожним змістовним модулем;
- оцінювання виконання лабораторних робіт;
- оцінювання тренінгів;
- оцінювання результатів самостійної роботи;
- підсумковий екзамен.

10. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100 – бальною шкалою) з дисципліни «Програмування П» визначається як середньозважена величина, в залежності від питомої ваги кожної складової залікового кредиту %:

Модуль 1		Модуль 2		Модуль 3	Модуль 4	Модуль 5
10%	10%	10%	10%	5%	15%	40%
Поточне оцінювання	Модульний контроль 1	Поточне оцінювання	Модульний контроль 2	Тренінги	Самостійна робота	Екзамен
Визначається як середнє арифметичне за захист лабораторних робіт №1-6.	Підсумкова письмова робота за темами №1-7.	Визначається як середнє арифметичне за захист лабораторних робіт №7-12.	Підсумкова письмова робота за темами № 8-15.	Визначається як середнє арифметичне з оцінок за виконання та презентацію 1 завдання тренінгу.	Визначається як середнє арифметичне за виконання та презентацію 1 завдання самостійної роботи.	Теоретичні питання: 2 питання по 30 балів - мах 60 балів. Практичне завдання - мах 40 балів

Шкала оцінювання:

За шкалою університету	За національною шкалою	За шкалою ECTS
90-100	відмінно	A (відмінно)
85-89	добре	B (дуже добре)
75-84		C (добре)
65-74	задовільно	D (задовільно)
60-64		E (достатньо)
35-59	незадовільно	FX (незадовільно з можливістю повторного складання)
1-34		F (незадовільно з обов'язковим повторним курсом)

11. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1	Мультимедійний проектор та проєкційний екран	1 -15
2	Персональні комп'ютери	1 -15
3	Комунікаційне програмне забезпечення (Zoom) для проведення занять у режимі он-лайн (за необхідності)	1 -15
4	Комунікаційна навчальна платформа (Moodle) для організації дистанційного навчання (за необхідності)	1 -15
5	Наявність доступу до мережі Інтернет	1 -15
6	Microsoft Windows, Microsoft Office 2013, Mozilla Firefox, FoxitReader, WinZip, DjVu Viewer, Total Commander, SQL Server 2015 Express, Visual Studio Code, Comuniti, Python IDE. PyCharm	1-15

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Aline Anthony. Python Programming: A Step By Step Guide From Beginner to Advanced (Beginner & Advanced). Independently published, 2020. — 173 p.
2. AloorRavi S. Metaprogramming with Python. Packt , 2022. — 402 p..
3. Campesato Oswald. Python for Programmers. Mercury Learning & Information, 2022. — 293 p.
4. Frandsen Edward. Python Programming: A Beginner's Guide To Become A Master Python Programmer. Edward Frandsen, 2021. — 142 p.
5. Amos David et al. Python Basics: A Practical Introduction to Python 3. 4th Edition. — David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler. — Real Python, 2021. — 643 p.
6. Balti Haythem. Job Ready Python. John Wiley & Sons, 2022. — 714 p.
7. Beazley David M. Python Distilled. Pearson, 2022. — 352 p.
8. Behrman Kennedy. Foundational Python for Data Science. Addison-Wesley/Pearson Education, 2021. — 256 p.
9. Brian O. Python Testing with pytest: Simple, Rapid, Effective, and Scalable. 2nd Edition. — Pragmatic Bookshelf, 2022. — 274 p.
10. Gaddis Tony. Starting out with Python. 5th Edition, Global Edition. — Pearson, 2022. — 895 p.
11. David Michael. Hand-On Python Tutorial: Project-Based Introduction to Programming For Beginners and Professionals. Independently published, 2021. — 208 p.
12. Mastrodomenico Rob. The Python Book. Wiley, 2022. — 258 p.