

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Волинський національний університет імені Лесі Українки
Факультет інформаційних технологій і математики
Кафедра комп'ютерних наук та кібербезпеки

СИЛАБУС
нормативного освітнього компонента
ПРОГРАМУВАННЯ
підготовки здобувачів освіти
першого (бакалаврського) рівня
спеціальності F5 Кібербезпека та захист інформації
освітньо-професійної програми
Кібербезпека та захист інформації

Силабус освітнього компонента «Програмування» підготовки бакалавра, галузі знань F Інформаційні технології, спеціальності F5 Кібербезпека та захист інформації, за освітньою програмою Кібербезпека та захист інформації.

Розробники:

Гришанович Т. О., доцент кафедри комп'ютерних наук та кібербезпеки, к.ф.-м.н., доцент.

Глинчук Л. Я., доцент кафедри комп'ютерних наук та кібербезпеки, к.ф.-м.н., доцент.

Погоджено

Гарантосвітньо-професійної

програми:



Черняшук Н.Л.

Силабус освітнього компонента затверджено на засіданні кафедри комп'ютерних наук та кібербезпеки

протокол № 2 від 17.09.2025 р.

Завідувач кафедри:



Гришанович Т. О.

I. Опис освітнього компонента

Найменування показників	Галузь знань, спеціальність, освітня програма, рівень освіти	Характеристика освітнього компонента
Денна форма здобуття освіти	F Інформаційні технології F5 Кібербезпека та захист інформації Кібербезпека та захист інформації бакалавр	Нормативна
Кількість годин/кредитів 330 / 11		Рік навчання 1
		Семестр 1,2-ий
		Лекції 64 год.
ІНДЗ: немає		Лабораторні 110 год.
		Самостійна робота 134 год.
		Консультації 22 год.
	Форма контролю: екзамен, екзамен	
Мова навчання: українська		

II. Інформація про викладача ППП

Глинчук Людмила Ярославівна

Науковий ступінь кандидат фізико-математичних наук

Вчене звання доцент

Посада доцент кафедри комп'ютерних наук та кібербезпеки

Контактна інформація hlynchuk.ludmila@vnu.edu.ua

Дні занять <https://ps.vnu.edu.ua/cgi-bin/timetable.cgi?n=700>

ППП Гришанович Тетяна Олександрівна

Науковий ступінь кандидат фізико-математичних наук

Вчене звання доцент

Посада доцент кафедри комп'ютерних наук та кібербезпеки

Контактна інформація hryshanovych.tatiana@vnu.edu.ua

Дні занять <https://ps.vnu.edu.ua/cgi-bin/timetable.cgi?n=700>

III. Опис освітнього компонента

1. Анотація освітнього компонента.

Освітній компонент «Програмування» відноситься до переліку дисциплін циклу професійної підготовки. Навчальна дисципліна присвячена сучасним технологіям програмування: структурного, об'єктно-орієнтованого, подійно-орієнтованого. Дисципліна покликана формувати у здобувача знання та вміння щодо мов програмування C++, Python, виробити техніку програмування для подальшого застосування при розробці програм, ознайомити із патернами проектування, із сервісом для спільної розробки програмного забезпечення GitHub та розробкою найпростіших unit-тестів.

2. Пререквізити: Знання та вміння із програмування на рівні програми ПЗСО.

Постреквізити: Теорія інформації і кодування, Криптографічний захист інформації, Технології веброзробки та захист вебресурсів, Практика з програмування, Курсова робота з програмування.

3. Мета і завдання освітнього компонента: метою дисципліни «Програмування» є вироблення у студентів навичок розробки програм із використанням різних парадигм програмування та мов, що підтримують ці парадигми. За результатами вивчення освітнього компонента здобувач повинен вміти реалізовувати алгоритми розв'язування задач в галузі інформаційних технологій, зокрема з кібербезпеки та захисту інформації, самостійно вибираючи парадигму програмування та використовуючи засоби відповідних мов програмування. Завданнями дисципліни «Програмування» є формування у здобувачів знань про етапи розробки програм; про редагування, трансляцію, компонування програм; про методи структурного, об'єктно-орієнтованого, подійно-орієнтованого програмування; про вбудовані типи даних у мовах програмування C++, Python, про паттерни проектування програмних систем, їх застосування до розробки програмного забезпечення, про базові поняття тестування програмного забезпечення. Також до завдань відноситься формування у здобувачів вмінь реалізовувати алгоритмічні конструкції, які підтримуються конкретними мовами програмування, вмінь реалізовувати алгоритм розв'язання задачі із застосуванням засобів мов програмування, вмінь реалізовувати типи даних користувача засобами конкретних мов програмування, розробляти графічний інтерфейс користувача, застосовувати на практиці патерни проектування, розробляти unit-тести.

4. Компетентності. Програмні результати навчання. Soft skills.

Загальні компетентності

ЗК 1. Здатність застосовувати знання у практичних ситуаціях.

ЗК 2. Знання та розуміння предметної області та розуміння професії.

ЗК 4. Вміння виявляти, ставити та вирішувати проблеми за професійним спрямуванням.

ЗК 5. Здатність до пошуку, оброблення та аналізу інформації.

ЗК 8. Здатність ухвалювати рішення та діяти, дотримуючись принципу неприпустимості корупції та будь яких інших проявів недоброчесності.

Спеціальні (фахові, предметні) компетентності

ФК 2. Здатність до використання інформаційно-комунікаційних технологій, сучасних методів і моделей інформаційної безпеки.

ПРН 2. Організовувати власну професійну діяльність, обирати оптимальні методи та способи розв'язування складних спеціалізованих задач та практичних проблем у професійній діяльності, оцінювати їхню ефективність.

ПРН 3. Використовувати результати самостійного пошуку, аналізу та синтезу інформації з різних джерел для ефективного рішення спеціалізованих задач професійної діяльності.

ПРН 5. Адаптуватися в умовах часткої зміни технологій професійної діяльності, прогнозувати кінцевий результат.

ПРН 6. Критично осмислювати основні теорії, принципи, методи і поняття у навчанні та професійній діяльності.

ПРН 15. Використовувати сучасне програмно-апаратне забезпечення інформаційно-комунікаційних технологій.

ПРН 53. Вирішувати задачі аналізу програмного коду на наявність можливих загроз.

Soft skills

Критичне мислення та аналітичні здібності. Уміння розкласти складну задачу на підзадачі. Логічний підхід до пошуку помилок і оптимізації коду.

Системне мислення. Розуміння взаємозв'язку між частинами програми та всією системою. Формування навички бачити програму як цілісний механізм.

Навички вирішення проблем. Розробка ефективних алгоритмів. Гнучкість у виборі підходів для досягнення результату.

Уважність до деталей. Точність при написанні синтаксису. Вміння перевіряти вхідні дані та обробляти виняткові ситуації.

Комунікація. Вміння пояснити свій код іншим (усно, у коментарях чи документації); постановка запитань викладачу/команді; участь у code review.

Самоорганізація і тайм-менеджмент: планування роботи над програмою поетапно; розподіл часу між написанням, тестуванням, налагодженням; здатність працювати над кількома проектами.

Командна робота. Робота з колегами у спільних проектах (Git, командне програмування); узгодження стилю коду та підходів; прийняття конструктивної критики.

Навички навчання (self-learning). Здатність швидко освоювати нові мови, бібліотеки й технології; розвиток звички постійно оновлювати знання; вміння працювати з документацією та сторонніми джерелами.

5. Структура освітнього компонента.

Семестр 1

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/ Бали
Змістовий модуль 1. Вступ до програмування. Мова програмування C++.						
Тема 1. Поняття задачі, алгоритму, програми. Стратегії розробки алгоритмів.	4	2		2		ДС
Тема 2. Основні парадигми програмування. Класифікація мов програмування. Структура мови програмування.	4	2		2		ДС

Тема 3. Вступ до мови C++. Директиви препроцесора. Структура програми.	6	2	2	2		Звіт по лаб. роботі/2
Тема 4. Класифікація структур даних. Типи даних в мові C++.	6	2	2	2		Звіт по лаб. роботі/2
Тема 5. Базові алгоритмічні конструкції. Їх реалізація мовою програмування.	12	2	6	2	2	Звіт по лаб. роботі/2
Тема 6. Функції. Перевантаження функцій.	8	2	4	2		Звіт по лаб. роботі/2
Тема 7. Рекурентні співвідношення та рекурсія.	10	2	2	4	2	Звіт по лаб. роботі/2
Тема 8. Адресація в мові C++. Показчики. Система динамічного розподілу пам'яті.	10	2	2	4	2	Звіт по лаб. роботі/2
Разом за модулем 1	60	16	18	20	6	18
Змістовий модуль 2. Алгоритми та структури даних						
Тема 1. Масиви. Особливості реалізації програм із використанням масивів	12	2	6	4		Звіт по лаб. роботі/9
Тема 2. Методи сортування та пошуку числових даних.	10	2	4	4		Звіт по лаб. роботі/6
Тема 3. Рядки. Особливості роботи з рядками. Методи пошуку підрядка в рядку. Генерування текстів.	12	2	6	4		Звіт по лаб. роботі/9
Тема 4. Структури (записи). Особливості реалізації програм із використанням структур (записів) та масивів структур (записів).	10	2	4	4		Звіт по лаб. роботі/6
Тема 5. Файли. Робота з файлами в C++.	8	2	2	4		Звіт по лаб. роботі/3
Тема 6. Динамічні структури даних.	8	2	2	4		Звіт по лаб. роботі/3
Тема 7. Хеш-функції. Хеш-таблиці.	8	2	2	4		Звіт по лаб. роботі/4
Тема 8. Графи. Алгоритми на графах.	12	2	4	4	2	Звіт по лаб. роботі/8
Тема 9. Дерева. Бінарні дерева. Обхід дерев.	10	2	2	4	2	Звіт по лаб. роботі/4
Разом за модулем 2	90	18	32	36	4	52
Модульна контрольна робота 1						15
Модульна контрольна робота 2						15

Всього годин/Балів	150	34	50	56	10	100
---------------------------	-----	----	----	----	----	-----

Семестр 2

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/ Бали
Змістовий модуль 3. Вступ до програмування мовою Python						
Тема 1. Вступ до мови програмування Python. Основи мови Python (синтаксис, змінні, типи даних). Структура програми.	6	2	2	2		Звіт по лаб. роботі/2
Тема 2. Числові дані. Винятки та їх обробка. Реалізація базових алгоритмічних структур мовою Python.	10	2	4	4		Звіт по лаб. роботі/4
Тема 3. Структури даних в Python: списки, кортежі, словники, рядкові величини, множини.	11	2	4	4	1	Звіт по лаб. роботі/4
Тема 4. Функції, лямбда функції та рекурсія.	7	2	2	2	1	Звіт по лаб. роботі/3
Тема 5. Робота з файлами в Python.	8	2	2	4		Звіт по лаб. роботі/3
Тема 6. Модулі в Python. Створення unit-тестів для мови Python.	11	2	4	4	1	Звіт по лаб. роботі/6
Разом за модулем 3	53	12	18	20	3	22
Змістовий модуль 4. Об'єктно-орієнтоване програмування мовою Python						
Тема 1. Оголошення класу, реалізація класу мовою Python. Конструктор, деструктор. Реалізація інкапсуляції мовою Python.	8	2	2	4		Звіт по лаб. роботі/2
Тема 2. Сетери, гетери, делетери. Статичні поля та методи. Графічна бібліотека turtle та її застосування. Реалізація класу на базі уже створеного класу: композиція.	10	2	4	4		Звіт по лаб. роботі/4
Тема 3. Реалізація класу на базі уже створеного класу: наслідування, множинне наслідування, багаторівневе наслідування. Порядок виклику методів (MRO) в Python.	11	2	4	4	1	Звіт по лаб. роботі/2
Тема 4. Реалізація поліморфізму мовою Python. Віртуальний метод.	9	2	2	4	1	Звіт по лаб. роботі/4

Зв'язки між класами: асоціація, агрегація, композиція, спадкування, залежність, реалізація.						
Тема 5. Спеціальні поля та методи. Особливості перевантаження операторів.	9	2	2	4	1	Звіт по лаб. роботі/2
Разом за модулем 4	47	10	14	20	3	14
Змістовий модуль 5. Подійно-орієнтоване програмування у Python						
Тема 1. Створення графічних програм GUI за допомогою модуля Tkinter на мові програмування Python. Перша програма. Віджет Button (кнопка). Позиціювання: Pack, Place, Grid. Обробка подій.	10	2	4	4		Звіт по лаб. роботі/4
Тема 2. Особливості, властивості та робота з віджетами: Label, Frame, Text, Scrollbar, Entry, Radiobutton, Checkbutton	11	2	4	4	1	Звіт по лаб. роботі/6
Тема 3. Особливості, властивості та робота з віджетами: Scale, Listbox, Canvas, Menu, MessageBox, звичайні вікна та діалогові.	9		4	4	1	Звіт по лаб. роботі/6
Разом за модулем 5	30	4	12	12	2	14
Змістовий модуль 6. Програмування задач пов'язаних з кібербезпекою						
Тема 1. Переведення чисел з однієї системи числення в іншу за алгоритмом. Функції перетворення hex(), oct(), bin(), int() . Математичні задачі пов'язані з кібербезпекою.	6	2	2	2		Звіт по лаб. роботі/2
Тема 2. Найпростіше шифрування у Python: шифри заміни та перестановки. Додавання по модулю 2.	8		4	4		Звіт по лаб. роботі/6
Тема 3. Модулі Python для шифрування (зовнішні модулі: cryptography, PyCryptodome) та хешування, генерації випадкових ключів, архівування (внутрішні модулі).	7		2	4	1	Звіт по лаб. роботі/3
Тема 4. Аналіз програмного коду на наявність можливих загроз. Статичний аналіз та модуль ast.	7		2	4	1	Звіт по лаб. роботі/3
Разом за модулем 6	28	2	10	14	2	14

Змістовий модуль 7. Патерни проєктування						
Тема 1. Патерни проєктування: породжуючі патерни.	8	2	2	4		Звіт по лаб. роботі/2
Тема 3. Патерни проєктування: структурні патерни.	7		2	4	1	Звіт по лаб. роботі/2
Тема 3. Патерни проєктування: поведінкові патерни.	7		2	4	1	Звіт по лаб. роботі/2
Разом за модулем 7	22	2	6	12	2	6
Тестування						10
Модульна контрольна робота 1						10
Модульна контрольна робота 2						10
Всього годин/Балів	180	30	60	78	12	100
Всього	330	64	110	134	22	

Методи контролю*: ДС – дискусія, ДБ – дебати, Т – тести, ТР – тренінг, РЗ/К – розв’язування задач/кейсів, ІНДЗ/ІРС – індивідуальне завдання/індивідуальна робота здобувача освіти, РМГ – робота в малих групах, МКР/КР – модульна контрольна робота/ контрольна робота, Р – реферат, а також аналітична записка, аналітичне есе, аналіз твору тощо.

6. Завдання для самостійного опрацювання.

Самостійна робота здобувачів включає в себе:

Семестр 1

Опрацювання лекційного матеріалу.

17 год

Перевірка здійснюється під час лабораторних занять та оцінюється при виставленні оцінки за змістовий модуль.

Підготовка до лабораторних занять, виконання домашніх завдань.

25 год

Перевірка здійснюється під час практичних занять.

Систематизація вивченого матеріалу перед контрольними заходами, тестуванням.

4 год

Перевірка здійснюється під час контрольних заходів, тестування.

Вивчення тем, що не розглядаються в курсі лекцій.

10 год

Перевірка здійснюється під час модульних контрольних заходів і оцінюється відповідною кількістю балів.

№ з/п	Тема	Кількість годин
1	Сервіс для спільної розробки програмного забезпечення GitHub.	4
2	Засіб автодоповнення коду GitHub Copilot.	2
3	Засоби автоамічного генерування коду.	4

	Всього	10
--	---------------	----

Семестр 2

Опрацювання лекційного матеріалу.	10 год
Перевірка здійснюється під час лабораторних занять та оцінюється при виставленні оцінки за змістовий модуль.	
Підготовка до лабораторних занять, виконання домашніх завдань.	30 год
Перевірка здійснюється під час лабораторних занять.	
Систематизація вивченого матеріалу перед контрольними заходами, тестуванням.	6 год
Перевірка здійснюється під час контрольних заходів, тестування.	
Вивчення тем, що не розглядаються в курсі лекцій.	32 год
Перевірка здійснюється під час лабораторних робіт, модульних контрольних заходів і оцінюється відповідною кількістю балів.	

№ з/п	Тема	Кількість годин
1	Вбудовані модулі у Python: datetime, time, date formatting,calendar, collections, os, pprint, re, sys та інші.	4
2	Принципи узагальненого програмування в Python.	4
3	Абстрактні класи в мові Python. Поняття про метапрограмування. Декоратори у мові Python.	4
4	Ітератори та генератори. Інтерфейси у мові Python.	4
5	Порівняння процедурного програмування та ООП.	4
6	Стилізація графічних програм GUI за допомогою модуля Tkinter.	4
7	Зовнішні модулі Python для програмування задач кібербезпеки.	4
8	Аналіз програмного коду на наявність можливих загроз. Динамічний аналіз у Python	4
	Всього	32

IV. Політика оцінювання

Політика викладача щодо студента

Усі учасники освітнього процесу повинні дотримуватись вимог чинного законодавства України, Статуту і Правил внутрішнього розпорядку ВНУ імені Лесі Українки, загальноприйнятих моральних принципів, правил поведінки та корпоративної культури; підтримувати атмосферу доброзичливості, відповідальності, порядності й толерантності. Атмосфера на заняттях повинна бути творчою, відкритою до конструктивної критики. Недопустимі запізнення на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття; списування. Очікується, що всі студенти

відвідають усі лекції і лабораторні заняття курсу. Кожен студент повинен бути учасником дистанційних курсів Програмування 1 (<https://moodle-cs.vnu.edu.ua/course/view.php?id=217>), Програмування 2, розміщених на платформах дистанційного навчання Moodle. Завдання для практичного виконання (лабораторні роботи, ІНДЗ, самостійні роботи), завдання підсумкового контролю (тести, контрольні роботи, що передбачають розробку програм) здаються із використанням засобів дистанційного курсу.

Політика щодо академічної доброчесності

Під час навчання учасники освітнього процесу зобов'язані дотримуватися академічної доброчесності: етичних принципів та визначених законом правил, якими мають керуватися учасники освітнього процесу під час навчання, викладання та провадження наукової діяльності.

Дотримання академічної доброчесності здобувачами передбачає: самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю (для осіб з особливим освітніми потребами ця вимога застосовується з урахуванням їх індивідуальних потреб і можливостей); посилення на джерела інформації у разі використання ідей, тверджень, відомостей; дотримання норм законодавства про авторське право; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності.

Під час оцінювання результатів навчання студенти не користуються забороненими засобами (мобільний телефон, планшет, конспект, навчальна література, інші джерела інформації, в тому числі Інтернет-ресурси), самостійно виконують запропоновані завдання. При виконанні лабораторних робіт з курсу здобувачі мають право використовувати власні ноутбуки, якщо вони підтримують необхідне програмне забезпечення.

Політика щодо дедлайнів та перескладання

Якщо здобувач вищої освіти був відсутній на заняттях з будь-якої причини, він/вона вивчають теоретичний матеріал самостійно використовуючи навчальні посібники, конспекти лекцій, матеріали дистанційних курсів Програмування 1 (<https://moodle-cs.vnu.edu.ua/course/view.php?id=217>), Програмування 2, розміщених на платформі дистанційного навчання Moodle, виконують всі домашні завдання. Прозвітуватися про виконання завдань можна, використовуючи вищезазначені дистанційні курси, або під час консультацій, одночасно при цьому з'ясувати незрозумілі моменти, задати запитання викладачу. Існує можливість використання форуму дистанційного курсу. Перескладання контрольних робіт та тестувань заборонено.

Можливість визнання результатів навчання, отриманих у формальній, неформальній та інформальній освіті

Відповідно до пункту 3.3 Положення про визнання результатів навчання, отриманих у формальній, неформальній та/або інформальній освіті у Волинському національному університеті імені Лесі Українки з дисципліни Програмування у першому семестрі визнання результатів навчання не проводиться.

У другому семестрі існує можливість перезарахування деяких тем. Тобто, якщо здобувач самостійно набув компетентностей із програмування мовою Python, використовуючи дистанційний курс «Програмування для всіх: основи Python» (<https://prometheus.org.ua/prometheus-free/python-fundamentals-for-everyone/>) та отримав

сертифікат про завершення курсу, йому буде перезараховано Теми 1, 2, 4, модуля 5 із результатом 9 балів, відповідно – з дистанційного курсу «Python: Структури даних» (<https://prometheus.org.ua/prometheus-free/python-data-structures/>), буде перезараховано Теми 3, 5, модуля 5, із результатом 7 балів. Якщо здобувач самостійно набув компетентностей, використовуючи дистанційний курс Programming in Python (<https://www.coursera.org/learn/programming-in-python/home/week/1>) та отримав сертифікат про завершення курсу, йому буде перезараховано змістовий модуль 5 та 6 із результатом 36 балів.

Можливість отримати додаткові (бонусні) бали.

Відповідно до пункту 4.5 Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Волинського національного університету імені Лесі Українки здобувачам освіти, які брали участь у роботі конференцій, підготовці наукових публікацій, в олімпіадах, конкурсах студентських наукових робіт, спортивних змаганнях, мистецьких конкурсах тощо й досягли значних результатів, може бути присуджено додаткові (бонусні) бали, які зараховуються як результати поточного контролю з відповідного ОК. Систему бонусних балів погоджує науково-методична комісія факультету (інституту). При цьому загальна кількість балів за поточну роботу, у випадку ОК, де передбачено екзамен, не може перевищувати 70 балів.

Здобувачі освіти мають можливість отримати додаткові бали з ОК «Програмування» за умови участі у заходах, передбачених документом: [Витяг НМК №1](#).

V. Підсумковий контроль

Підсумковою формою контролю освітнього компонента «Програмування» є іспити в обох семестрах.

Оцінювання навчальних досягнень здійснюється за 100 бальною шкалою. Оцінка включає в себе поточний контроль (оцінюється робота на парах, вчасне і якісне виконання домашніх завдань) та підсумковий контроль (самостійне виконання індивідуальних завдань, контрольні роботи, перевірка теоретичної підготовки у формі тестування, ІНДЗ). Максимальна кількість балів, яку може отримати здобувач під час поточного оцінювання за семестр – 70 балів. Максимальна кількість балів, яку може отримати здобувач за підсумковий контроль за семестр складає 30 балів.

Протягом поточної роботи здобувач освіти має набрати не менше 35 балів (як допуск до складання іспиту). Для отримання допуску здобувач освіти має відпрацювати лабораторні роботи до дати іспиту під час основної сесії. Оцінки за відпрацьовані роботи викладач виставляє в електронний журнал успішності поруч або замість «н». У випадку, якщо здобувач освіти отримав менше, ніж 35 балів, він не може бути допущеним до екзамену і повинен бути відрхований за академічну неуспішність.

Якщо здобувач освіти успішно виконав усі завдання, передбачені силабусом, і набрав при цьому не менше як 75 балів, він може отримати підсумкову семестрову оцінку без складання іспиту. Така оцінка виставляється в день проведення екзамену в присутності здобувача освіти.

Семестр 1

Питання для іспиту

1. Дайте визначення алгоритму та назвіть його основні властивості.

2. Які основні етапи розв'язування задачі на комп'ютері?
3. Які стратегії розробки алгоритмів ви знаєте?
4. Наведіть приклад алгоритму з розгалуженням або циклом.
5. Назвіть основні парадигми програмування та наведіть приклади мов для кожної.
6. Які критерії використовують для класифікації мов програмування?
7. У чому полягає різниця між компільованими та інтерпретованими мовами?
8. Що таке синтаксис та семантика мови програмування?
9. Які особливості мови програмування C++?
10. Яку роль відіграють директиви препроцесора?
11. Опишіть структуру типової програми на C++.
12. Яке призначення функції `main()`?
13. Що таке структура даних і як вона класифікується?
14. Назвіть основні типи даних, що підтримуються в C++.
15. Яка різниця між цілими та дійсними типами даних?
16. Що таке приведення типів?
17. Які базові алгоритмічні конструкції ви знаєте?
18. Як реалізується оператор розгалуження в C++?
19. Як працюють цикли `for`, `while` та `do...while`?
20. Як можна поєднувати конструкції керування у складних алгоритмах?
21. Що таке функція та яку роль вона відіграє в програмі?
22. Яка різниця між оголошенням та викликом функції?
23. Поясніть механізм перевантаження функцій.
24. У чому відмінність між передачею параметрів за значенням і за посиланням?
25. Що таке рекурсія та які її властивості?
26. У чому різниця між прямою та непрямою рекурсією?
27. Які умови коректної роботи рекурсивної функції?
28. Наведіть приклад рекурсивного обчислення простої задачі.
29. Що таке покажчик і як його оголосити?
30. Як отримати адресу змінної в пам'яті?
31. Як виконується динамічний розподіл пам'яті?
32. Як використовуються покажчики у функціях?
33. Що таке масив і як він оголошується?
34. Як отримати доступ до елемента масиву?
35. Як виконати пошук мінімального або максимального елемента масиву?
36. Які особливості використання масивів у програмах?
37. Назвіть основні алгоритми сортування.
38. У чому полягає суть сортування бульбашкою?
39. Як працює бінарний пошук?
40. Порівняйте ефективність лінійного та бінарного пошуку.
41. Що таке рядок у C++ і як він зберігається?
42. Як визначити довжину рядка?
43. Які методи пошуку підрядка ви знаєте?
44. Як можна згенерувати випадковий текст?
45. Що таке структура (`struct`) у програмуванні?
46. Як звертатися до полів структури?
47. Як організувати масив структур?
48. Які переваги використання структур у програмах?
49. Які види файлів ви знаєте?
50. Як відкрити та закрити файл у C++?
51. Як здійснюється читання та запис даних у файл?
52. У чому різниця між текстовими та бінарними файлами?

53. Що таке динамічні структури даних?
54. Як реалізується список у C++?
55. Що таке стек і черга?
56. Які переваги динамічних структур перед статичними?
57. Що таке хеш-функція та які її властивості?
58. Як організована робота хеш-таблиці?
59. Що таке колізія та як її можна вирішити?
60. Які переваги пошуку в хеш-таблицях?
61. Що таке граф та які його види існують?
62. Як графи можна представити у програмі?
63. Які алгоритми обходу графів ви знаєте?
64. У чому суть алгоритму пошуку в ширину та в глибину?
65. Що таке дерево та які його основні властивості?
66. Що таке бінарне дерево?
67. Які способи обходу бінарного дерева ви знаєте?
68. Наведіть приклади практичного застосування дерев.

Приклад завдання для іспиту:

Задано матрицю $A[m, n]$. В останньому рядку та останньому стовпцеві матриці визначити присутність цілого числа x та його індекс шляхом застосування алгоритму двійкового (бінарного) пошуку, якщо елементи цих рядка та стовпця впорядковані у порядку неспадання. Для реалізації використати засоби мови програмування C++.

Семестр 2

Питання для іспиту

1. Вступ до мови програмування Python. Основи мови Python (синтаксис, змінні, типи даних). Структура програми.
2. Числові дані. Винятки та їх обробка.
3. Реалізація базових алгоритмічних структур мовою Python.
4. Структури даних у Python: списки, кортежі.
5. Структури даних у Python: словники, рядкові величини, множини.
6. Функції, лямбда функції та рекурсія.
7. Робота з файлами в Python.
8. Модулі в Python.
9. Створення unit-тестів для мови Python.
10. Вбудовані модулі у Python: `datetime`, `time`, `date formatting`, `calendar`, `collections`, `os`, `pprint`, `re`, `sys` та інші.
11. Принципи узагальненого програмування в Python.
12. Оголошення класу, реалізація класу мовою Python. Конструктор, деструктор.
13. Реалізація інкапсуляції мовою Python.
14. Сетери, гетери, делетери. Статичні поля та методи. Графічна бібліотека `turtle` та її застосування.
15. Реалізація класу на базі уже створеного класу: композиція.
16. Реалізація класу на базі уже створеного класу: наслідування, множинне наслідування.
17. Реалізація класу на базі уже створеного класу: багаторівневе наслідування. Порядок виклику методів (MRO) в Python.
18. Реалізація поліморфізму мовою Python. Віртуальний метод.

19. Зв'язки між класами: асоціація, агрегація, композиція, спадкування, залежність, реалізація.
20. Спеціальні поля та методи. Особливості перевантаження операторів.
21. Абстрактні класи в мові Python.
22. Поняття про метапрограмування. Декоратори у мові Python.
23. Ітератори та генератори. Інтерфейси у мові Python.
24. Створення графічних програм GUI за допомогою модуля Tkinter на мові програмування Python. Перша програма. Віджет Button (кнопка).
25. Позиціонування: Pack, Place, Grid. Обробка подій.
26. Особливості, властивості та робота з віджетами: Label, Frame, Text, Scrollbar.
27. Особливості, властивості та робота з віджетами: Entry, Radiobutton, Checkbutton.
28. Особливості, властивості та робота з віджетами: Scale, Listbox, Canvas.
29. Особливості, властивості та робота з віджетами: Menu, MessageBox, звичайні вікна та діалогові.
30. Стилiзація графічних програм GUI за допомогою модуля Tkinter.
31. Переведення чисел з однієї системи числення в іншу за алгоритмом. Функції перетворення hex(), oct(), bin(), int(). Математичні задачі пов'язані з кібербезпекою. Приклади.
32. Найпростіше шифрування у Python: шифри заміни. Приклади.
33. Найпростіше шифрування у Python: перестановки. Додавання по модулю 2. Приклади.
34. Модулі Python для шифрування (зовнішні модулі: cryptography, PyCryptodome) та хешування, генерації випадкових ключів, архівування (внутрішні модулі).
35. Аналіз програмного коду на наявність можливих загроз. Статичний аналіз та модуль ast.
36. Зовнішні модулі Python для програмування задач кібербезпеки.
37. Аналіз програмного коду на наявність можливих загроз. Динамічний аналіз у Python.
38. Патерни проектування: породжуючі патерни.
39. Патерни проектування: структурні патерни.
40. Патерни проектування: поведінкові патерни.

Приклад завдання для іспиту:

1. Засобами мови програмування Python створити клас «Книга», який представлятиме інформацію про книгу (назва, автор, кількість сторінок). Використати механізми інкапсуляції для обмеження доступу до приватних атрибутів класу. Створити клас «ЕлектроннаКнига», який буде наслідувати клас «Книга», і додати до нього атрибуту для зберігання формату файлу та розміру в мегабайтах. Використати механізм поліморфізму для перевизначення методу, що виводить інформацію про книгу. Також створити клас «Автор», який міститиме інформацію про автора (ім'я, прізвище, кількість написаних книг). Використати композицію, зв'язавши клас «Автор» з класом «Книга», щоб кожна книга мала відповідного автора.
2. Створити міні-проект для керування списком завдань (To-Do List) за допомогою модуля Tkinter. У програмі повинні бути реалізовані функціональні елементи для введення, редагування, перегляду та видалення завдань. Програма використовує різноманітні

віджети для взаємодії з користувачем і позиціонування елементів за допомогою різних методів.

Шкала оцінювання

Шкала оцінювання знань здобувачів освіти з освітніх компонентів, де формою контролю є іспит

Оцінка в балах	Лінгвістична оцінка	Оцінка за шкалою ECTS	
		оцінка	пояснення
90–100	Відмінно	A	відмінне виконання
82–89	Дуже добре	B	вище середнього рівня
75–81	Добре	C	загалом хороша робота
67–74	Задовільно	D	непогано
60–66	Достатньо	E	виконання відповідає мінімальним критеріям
1–59	Незадовільно	Fx	необхідне перескладання

VI. Рекомендована література та інтернет-ресурси

Семестр 1

1. Бородін В. А. Мова програмування C++.: навч. посіб. К. : Видавничо-поліграфічний центр "Київський університет", 2024. 277 с.
2. Гришанович Т. О. Програмування 1 [Електронний ресурс] : електронний курс навчальної дисципліни, затверджений НМР ВНУ імені Лесі Українки, протокол № 7 від 19.03.2025. ВНУ ім. Лесі Українки, 2025. URL: <http://cs.vnu.edu.ua/moodle/course/view.php?id=217>
3. Зеленський О.С., Лисенко В.С. Основи програмування на C++: навчальний посібник. Кривий Ріг: Державний університет економіки і технологій, 2023. 269 с.
4. Ковалюк Т. В. Алгоритмізація та програмування: Підручник. Львів: «Магнолія 2006», 2024. 400 с.
5. Програмування: підручник [Електронний ресурс] / укладач Л. Я. Глинчук, Т. О. Гришанович; ВНУ ім. Лесі Українки. Електронні текстові дані (1 файл: 3 201 КБ). Луцьк: ВНУ ім. Лесі Українки, 2022. 160 с. URI: <https://evnuir.vnu.edu.ua/handle/123456789/2064>
6. Прокопенко О. В. Мова програмування C/C++. Практикум: навчальний посібник / О. В.

Семестр 2

1. Глинчук Л.Я. Застосування Python у наукових дослідженнях. *Сучасна наука та освіта Волині*: матеріали всеук. наук.-практ. конференції студентів, аспірантів та молодих вчених. (Луцьк, 22 травня 2025 р.) Луцьк, ФОП Мажула Ю.М., 2025 р. С. 166-167 URL: <https://surl.lj/wlslgk>
2. Глинчук Л.Я. Програмування 2 [Електронний ресурс]: електронний курс навчальної дисципліни, URL: <https://moodle.vnu.edu.ua/course/view.php?id=3098>
3. Глинчук Л.Я. Мультипотоківість у Python як інструмент підвищення ефективності програм. *Проблеми комп'ютерних наук, програмного моделювання та безпеки цифрових систем*: тези доповідей II міжн. наук.-практ. конф. (Луцьк-Світязь, 9-11 червня 2025 р.). URL: <https://apcssm.vnu.edu.ua/index.php/conf/article/view/159/122>
4. Глинчук Л.Я. Засоби мови програмування Python для вирішення задач кібербезпеки та приклади їх використання. *Інформаційні технології і автоматизація – 2024*: матеріали XVII міжн. наук.-практ. конференції. (Одеса, 31 жовтня - 1 листопада 2024 р.). Одеса, Видавництво ОНТУ, 2024 р. С.181-184.
5. Глинчук Л.Я. Програмування задачі на композицію конструкцій мовою Python. *Математика. Інформаційні технології. Освіта*: матеріали XII міжн. наук.-практ. конф. (Луцьк-Світязь, 3-5 червня 2023 р.). Луцьк: Волинський національний університет імені Лесі Українки, 2023. С.67-69.
6. Костюченко А. О. Основи програмування мовою Python навчальний посібник. Чернігів : ФОП Баликіна С.М., 2020. 176 с.
7. Татарчук Д. Д., Діденко Ю. В. Об'єктно-орієнтоване програмування мовою Python конспект лекцій. Київ : КПП ім. Ігоря Сікорського, 2021. 128 с.
8. Юрченко І. В., Сікора В. С. Програмування мовою Python: навчальний посібник. Чернівці : Чернів. нац. ун-т. 2022. 104 с.
9. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник. Київ: ВПЦ «Київський Університет», 2020. 152 с.
10. Фрімен Е. Head First. Патерни проектування. Київ : Фабула, 2020. 672 с.
11. Процедурне програмування VS ООП | FoxmindED. FoxmindEd. URL: <https://foxminded.ua/protsedurne-prohramuvannia/>
12. Уроки Python - aCode. aCode. URL: <https://acode.com.ua/lessons-python/>
13. Python | BestProg. BestProg | Програмування: теорія та практика. URL: <https://www.bestprog.net/uk/sitemap ua/python-ua/>
14. Python – просто!. Google Sites: Sign-in. URL: <https://sites.google.com/comp-sc.if.ua/python-easy/домашня-сторінка>
15. Patterns. Патерни | BestProg. BestProg | Програмування: теорія та практика. URL: <https://www.bestprog.net/uk/паттерни/>
16. Python – Tkinter. Інформатика – це просто. Bevor Sie zu YouTube weitergehen. URL: <https://www.youtube.com/playlist?list=PLYRayxBfqmqFWkGHLsbgcARvYdwH3nko>

17. Бібліотека Tkinter мови Python. URL:
<https://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/tkinter/index.html#Text>
18. Python у профільному рівні - Системи числення. *Google Sites: Sign-in*.
URL: <https://sites.google.com/view/python-11-profilniy/методи-проектування-алгоритмів/системи-числення>
19. Python: Проєкт "Шифрування" (EasyGUI) | Інформатика - ДистОсвіта. *На головну | Інформатика - ДистОсвіта*. URL: <https://dystosvita.org.ua/mod/page/view.php?id=748>