

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 122 Комп'ютерні науки
освітньо-професійної програми
Комп'ютерні науки та інформаційні технології

Силабус освітнього компонента «Програмування» підготовки бакалавра, галузі знань 12 Інформаційні технології, спеціальності 122 Комп'ютерні науки, за освітньою програмою Комп'ютерні науки та інформаційні технології.

Розробники:

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

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

Погоджено

Гарант освітньо-професійної програми:



Гришанович Т. О.

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, освітньо-професійна /освітньо-наукова/освітньо-творча програма, освітній рівень	Характеристика освітнього компонента
Денна форма здобуття освіти	12 Інформаційні технології 122 Комп'ютерні науки Комп'ютерні науки та інформаційні технології, Бакалавр	Нормативна
Кількість годин/кредитів 420 / 14		Рік навчання 1-2
		Семестр 1-3-ий
ІНДЗ: є		Лекції 86 год.
		Лабораторні 150 год.
		Самостійна робота 160 год.
		Консультації 24 год.
	Форма контролю: залік, 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 <http://194.44.187.20/>

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 <http://194.44.187.20/>

III. Опис освітнього компонента**1. Анотація ОК.**

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++, Java, Python, виробити техніку програмування для подальшого застосування при розробці програм, ознайомити із сервісом для спільної розробки програмного забезпечення GitHub, із засобами автоматичної генерації програмного коду та розробкою найпростіших unit-тестів.

2. Пререквізити.

Освітній компонент і так починається з I семестру, тому попередні ОК відсутні.

Постреквізити.

ОК «Спеціалізовані мови програмування», «Програмування та підтримка вебзастосовань», «Системне програмування та операційні системи» та ін.

2. Мета і завдання освітнього компонента: метою ОК «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 комп'ютерних наук, самостійно вибираючи парадигму програмування та використовуючи засоби відповідних мов програмування. Завданнями ОК «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++, Java, Python, про базові поняття тестування програмного забезпечення. Також до завдань відноситься формування у здобувачів вмінь реалізовувати алгоритмічні конструкції, які підтримуються конкретними мовами програмування, вмінь правильного оформлення програмного коду, вмінь реалізовувати алгоритм розв'язання задачі із застосуванням засобів мов програмування, вмінь реалізовувати типи даних користувача засобами конкретних мов програмування, розробляти unit-тести, використовувати засоби для автоматичного генерування програмного коду.

3. Компетентності. Програмні результати навчання. Soft skills.

Загальні компетентності:

ЗК 1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК 2. Здатність застосовувати знання у практичних ситуаціях.

ЗК 3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК5. Здатність спілкуватися іноземною мовою.

ЗК 6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК 7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК 8. Здатність генерувати нові ідеї (креативність).

ЗК 10. Здатність бути критичним і самокритичним.

ЗК 11. Здатність приймати обґрунтовані рішення.

ЗК 12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

ЗК 13. Здатність діяти на основі етичних міркувань.

ЗК16. Здатність ухвалювати рішення та діяти, дотримуючись принципу неприпустимості корупції та будь-яких інших проявів недоброчесності.

Спеціальні компетентності:

СК 1. Здатність до математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтовування вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування

СК 8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

Результати навчання:

ПРН 1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та

синтезу інформації в предметній області комп'ютерних наук.

ПРН 9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

Soft skills:

Критичне мислення та аналітичні здібності – вміння аналізувати завдання, розбивати його на підзадачі, знаходити оптимальні рішення.

Навички вирішення проблем (Problem-solving) – здатність шукати різні підходи, виправляти помилки, працювати з невизначеністю.

Уважність до деталей – адже навіть одна помилка у коді може зламати всю програму.

Логічне та алгоритмічне мислення – розвиток структурованого підходу до роботи й ухвалення рішень.

Командна робота та комунікація – під час групових проєктів чи роботи з чужим кодом необхідно вміти пояснювати власні ідеї та розуміти чужі.

Тайм-менеджмент та самоорганізація – планування часу для написання та тестування програм, робота в дедлайнах.

Стресостійкість та наполегливість – подолання труднощів у процесі налагодження коду, здатність не здаватися при виникненні помилок.

Креативність – пошук нестандартних рішень та власний підхід до розробки програм.

Навички навчання впродовж життя (lifelong learning) – постійне оновлення знань, робота з документацією, освоєння нових мов і технологій.

4. 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 C++.						
Тема 1. Основні парадигми програмування. Класифікація мов програмування. Структура мови програмування.	6	2		4		ДС
Тема 2. Вступ до мови C++. Директиви препроцесора. Структура програми, типи даних.	12	2	4	4	2	Звіт по лаб. роботі/4
Тема 3. Реалізація базових алгоритмічних конструкцій мовою C++. Функції.	14	2	8	4		Звіт по лаб. роботі/8
Тема 4. Адресація в мові C++. Показники.	8	2	2	4		Звіт по лаб. роботі/2
Тема 5. Обробка виняткових ситуацій.	6		2	4		Звіт по лаб. роботі/2
Разом за модулем 1	46	8	16	20	2	16
Змістовий модуль 2. Складені типи в мові програмування C++						
Тема 1. Система для керування та контролю версіями Git.	12	2	2	6	2	Звіт по лаб. роботі/2
Тема 2. Масиви. Особливості реалізації програм із використанням масивів.	14	2	6	6		Звіт по лаб. роботі/6
Тема 3. Файли. Робота з файлами	10	2	4	4		Звіт по лаб.

в C++.						роботі/4
Тема 4. Система динамічного розподілу пам'яті.	10	2	2	4	2	Звіт по лаб. роботі/2
Тема 5. Рядки. Особливості реалізації програм із використанням текстових рядків.	10	2	4	4		Звіт по лаб. роботі/4
Тема 6. Структури. Особливості реалізації програм із використанням структур та масивів структур.	10	2	4	4		Звіт по лаб. роботі/4
Тема 7. Об'єднання та переліки. Особливості реалізації програм із використанням об'єднань та переліків.	8	2	2	4		Звіт по лаб. роботі/2
Разом за модулем 2	74	14	24	32	4	24
Тестування						20
Модульна контрольна робота 1						10
Модульна контрольна робота 2						10
ІНДЗ 1						10
ІНДЗ 2						10
Всього годин/Балів	120	22	40	52	6	100

Семестр 2

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/ Бали
Змістовий модуль 3. Об'єктно-орієнтоване програмування мовою C++						
Тема 1. Вступ. Основні поняття об'єктно-орієнтованого програмування. Класи. Екземпляри класів.	10	2	4	4		Звіт по лаб. роботі/2
Тема 2. Дані та методи класу. Досяжність. Реалізація інкапсуляції у мові C++.	8	2	2	4		Звіт по лаб. роботі/2
Тема 3. Конструктор та деструктор - особливі методи класі. Види конструкторів.	8	2	2	4		Звіт по лаб. роботі/2
Тема 4. Особливості роботи з об'єктами. Масиви об'єктів.	10	2	4	4		Звіт по лаб. роботі/2
Тема 5. Наслідування. Реалізація механізму наслідування у мові програмування C++. Види наслідування	10	2	4	4		Звіт по лаб. роботі/4
Тема 6. Друзі класу. Перевантаження операторів.	12	2	4	4	2	Звіт по лаб. роботі/2
Тема 7. Реалізація поліморфізму в мові програмування C++. Поліморфізм віртуальних методів.	10	2	2	4	2	Звіт по лаб. роботі/2

Тема 8. Абстрактні класи.	8	2	2	4		Звіт по лаб. роботі/2
Разом за модулем 3	76	16	24	32	4	20
Змістовий модуль 4. Об'єктно-орієнтоване програмування мовою Java						
Тема 1. Вступ до мови програмування Java. Основні поняття. Базовий синтаксис, структура програми.	6	2	2	2		Звіт по лаб. роботі/2
Тема 2. Типи даних в мові Java, реалізація базових алгоритмічних структур.	10	2	4	4		Звіт по лаб. роботі/2
Тема 3. Дані та методи класу. Досяжність. Реалізація інкапсуляції у мові Java.	10	4	4	2		Звіт по лаб. роботі/4
Тема 4. Масив об'єктів у Java: Як створити, ініціалізувати та використовувати.	6	2	2	2		Звіт по лаб. роботі/2
Тема 5. Наслідування в Java. Реалізація множинного наслідування в Java.	10	2	4	4		Звіт по лаб. роботі/2
Тема 6. Реалізація поліморфізму в мові Java.	8	4	2	2		Звіт по лаб. роботі/2
Тема 7. Створення unit-тестів в середовищі IntelliJ IDEA.	12	2	4	4	2	Звіт по лаб. роботі/6
Тема 8. Засоби автоматичної генерації коду.	10		4	4	2	
Разом за модулем 4	74	18	26	24	4	20
Тестування						20
Модульна контрольна робота 1						10
Модульна контрольна робота 2						10
ІНДЗ 1						10
ІНДЗ 2						10
Всього годин/Балів	150	34	50	56	8	100

Семестр 3

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/Бали
Змістовий модуль 5. 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	9	2	4	2	1	Звіт по лаб. роботі/4

Python.						
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.	8	2	4	2		Звіт по лаб. роботі/3
Тема 5. Робота з файлами в Python.	6	2	2	2		Звіт по лаб. роботі/2
Тема 6. Модулі в Python. Створення unit-тестів для мови Python.	11	2	6	2	1	Звіт по лаб. роботі/7
Разом за модулем 5	51	12	22	14	3	22
Змістовий модуль 6. Об'єктно-орієнтоване програмування мовою Python						
Тема 1. Порівняння процедурного програмування та ООП. Оголошення класу, реалізація класу мовою Python. Конструктор, деструктор. Реалізація інкапсуляції мовою Python.	6	2	2	2		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.	6	2	2	2		Звіт по лаб. роботі/3
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/5
Тема 4. Реалізація поліморфізму мовою Python. Віртуальний метод. Зв'язки між класами: асоціація, агрегація, композиція, спадкування, залежність, реалізація.	9	2	2	4	1	Звіт по лаб. роботі/3
Тема 5. Спеціальні поля та методи. Особливості переважання операторів.	8	2	2	4		Звіт по лаб. роботі/2
Тема 6. Абстрактні класи. Поняття про метапрограмування. Декоратори.	9	2	4	4	1	Звіт по лаб. роботі/3
Разом за модулем 6	51	12	16	20	3	18
Змістовий модуль 7. Подійно-орієнтоване програмування у Python						
Тема 1. Створення графічних програм GUI за допомогою модуля Tkinter на мові програмування	9	2	4	2	1	Звіт по лаб. роботі/4

Python. Перша програма. Віджет Button (кнопка). Позиціонування: Pack, Place, Grid. Обробка подій.						
Тема 2. Особливості, властивості та робота з віджетами: Label, Frame, Text, Scrollbar, Entry, Radiobutton, Checkbutton	9	2	4	2	1	Звіт по лаб. роботі/4
Тема 3. Особливості, властивості та робота з віджетами: Scale, Listbox, Canvas, Menu, MessageBox, звичайні вікна та діалогові.	6		4	2		Звіт по лаб. роботі/4
Тема 4. Дизайн графічного інтерфейсу користувача: бібліотека PyQt. ПЗ QtDesigner.	17	2	8	6	1	Звіт по лаб. роботі/8
Презентація виконання індивідуального практичного завдання	9		2	6	1	Звіт/10
Разом за модулем 7	50	6	22	18	4	30
Тестування						10
Модульна контрольна робота 1						10
Модульна контрольна робота 2						10
Всього годин/Балів	152	30	60	52	10	100
Всього годин за курс	420	86	150	160	24	

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.

5. Завдання для самостійного опрацювання.

Самостійна робота здобувачів включає в себе:

Семестр 1

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. 20 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. 18 год

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

№ з/п	Тема	Кількість годин
1	Низхідне та висхідне проектування.	4
2	Модульне програмування.	4
3	Організація форматowanego виводу. Бібліотека <stdio>.	2
4	Бібліотеки динамічного компонування (DLL).	4
5	Особливості стандартів C11, C++11, C++14.	4
	Всього	18

Семестр 2

Опрацювання лекційного матеріалу. 8 год

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

Підготовка до лабораторних занять, виконання домашніх завдань. 18 год

Перевірка здійснюється під час лабораторних занять.

Систематизація вивченого матеріалу перед іспитом. 6 год

Перевірка здійснюється під час іспиту.

Вивчення тем, що не розглядаються в курсі лекцій. 24 год

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

№ з/п	Тема	Кількість годин
1	Ідентифікація та приведення типів в C++.	2
2	Види класів в C++.	6
3	Класи потокового введення-виведення.	4
4	Види класів в мові Java.	4
5	Засоби автоматичної генерації коду.	4
6	Мова ULM.	6
	Всього	24

Семестр 3

Опрацювання лекційного матеріалу. 15 год

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

Підготовка до лабораторних занять, виконання домашніх завдань. 15 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. 16 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.	2
2	Вбудовані модулі та функції для програмування задач, пов'язаних з кібербезпекою.	2
3	Принципи узагальненого програмування в Python.	2
4	Ітератори та генератори.	2
5	Інтерфейси у мові Python.	2
6	Рефлексія та інтроспекція у мові Python.	2
7	Бібліотеки Python для роботи з БД.	2
7	Стилізація графічних програм GUI за допомогою модуля Tkinter.	2
	Всього	16

IV. Політика оцінювання

Політика викладача щодо здобувача освіти

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, Програмування 2, Програмування 3, розміщених на платформах дистанційного навчання Moodle. (<https://moodle-cs.vnu.edu.ua/course/view.php?id=74>, <https://moodle-cs.vnu.edu.ua/course/view.php?id=85>, <https://moodle-cs.vnu.edu.ua/course/view.php?id=250>). 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); посилання 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», «Програмування 2», «Програмування 3», 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 або вживу, або через корпоративну пошту. Існує можливість використання форуму дистанційного курсу. Перескладання контрольних робіт та тестувань заборонено.

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

Під час вивчення освітнього компонента можливе визнання інших результатів навчання, отриманих у формальній, неформальній та/або інформальній освіті. Порядок визнання результатів навчання для здобувачів вищої освіти, набутих у: формальній освіті (академічна мобільність студентів на території України чи поза її межами, для студентів, які переводяться, поновлюються з інших ЗВО (вітчизняних чи іноземних); неформальній та/або інформальній освіті здійснюється згідно «ПОЛОЖЕННЯ про визнання результатів навчання, отриманих у формальній, неформальній та/або інформальній освіті у Волинському національному університеті імені Лесі Українки».

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» проводиться таким чином: 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 зараховано 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 зараховано Теми 3, 5, модуля 5, із результатом 6 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 зараховано змістовий модуль 5 та 6 із результатом 40 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.

І так, здобувачі освіти мають можливість отримати додаткові бали за вказаний вид робіт з ОК «Програмування» відповідно до таблиці витягу з протоколу № 1 засідання НМТ ФІТІМ ВНУ ім. Лесі Українки від 3.09.2025 р.

Системи бонусних балів для здобувачів освіти

Вид діяльності	Рівень / результат	Кількість бонусних балів
Студентські олімпіади	I місце	7
	II місце	5
	III місце	3
	Участь в олімпіаді	2
Конкурси студентських наукових робіт	Диплом I ступеня	7
	Диплом II ступеня	5
	Диплом III ступеня	3
Підготовка наукових публікацій	Публікація в WoS / Scopus	10
	Фахова стаття	7
	Нефахова стаття	5
	Публікація тез	2
Участь у конференціях	Виступ на конференції	2
Першість України з командного програмування	I місце	10
	II місце	8
	III місце	6
	Участь	4

V. Підсумковий контроль

Підсумковою формою контролю освітнього компонента «Програмування» є іспит у 2 та 3 семестрах та залік у 1 семестрі.

Для 1, 2 та 3 семестрів оцінювання здійснюється за 100-бальною шкалою. Оцінка включає в себе оцінювання всіх видів запланованої навчальної роботи протягом семестру: нараховується за якісне виконання лабораторних, контрольних, тестових робіт та виконання індивідуального завдання. Максимальна кількість балів, яку може отримати студент під час поточного оцінювання, у випадку заліку, за семестр – 100 балів.

Залік викладач виставляє за результатами поточної роботи за умови, що здобувач освіти виконав ті види навчальної роботи, які визначено силабусом ОК. У випадку, якщо здобувач освіти не відвідував окремі аудиторні заняття (з поважних причин), на консультаціях він має право відпрацювати пропущені заняття та добрати ту кількість балів, яку було визначено на пропущені теми. У дату складання заліку викладач записує у відомість суму поточних балів, які здобувач освіти набрав під час поточної роботи (шкала від 0 до 100 балів).

У випадку, якщо здобувач освіти протягом поточної роботи набрав менше як 60 балів, він складає залік під час ліквідації академічної заборгованості. У цьому випадку бали, набрані під час поточного оцінювання анулюються. Максимальна кількість балів на залік під час ліквідації академічної заборгованості, як правило, 100. У день складання заліку за основною сесією заборонено проводити додаткові опитування здобувача освіти, а також здобувач освіти не має права доздавати будь-який вид робіт, передбачений силабусом освітнього компоненту. На заліку, під час ліквідації академічної заборгованості, здобувач отримує комплексне завдання, яке охоплює всі теми і всі форми контролю, які пропонувалися при вивченні освітнього компонента.

Для 2 та 3 семестрів підсумковою формою контролю освітнього компонента є іспит. Оцінювання навчальних досягнень здійснюється за 100 бальною шкалою. Оцінка включає в себе поточний контроль та підсумковий модульний контроль. Максимальна кількість балів за поточний контроль з ОК, де форма контролю екзамен, становить 70 балів. Підсумковий модульний контроль здійснюється після завершення вивчення тем змістового модуля у формі

виконання здобувачем освіти модульного контрольного завдання (контрольної роботи, тесту, колоквиуму тощо) та проводиться або під час навчального заняття (його частини), або поза розкладом згідно окремого графіка. Максимальний бал, отриманий за модульні контрольні роботи, становить, як правило, не більше як 30 балів.

З ОК, де формою контролю є екзамен, протягом поточної роботи здобувач освіти має набрати не менше 35 балів (як допуск до складання іспиту). Для отримання допуску здобувач освіти має відпрацювати практичні або 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 відрахований за академічну неуспішність. Під час складання екзамену здобувач може отримати від 0 до 30 балів.

Якщо з ОК, де передбачено іспит, підсумкові бали поточного контролю та підсумкового модульного контролю становлять не менше як 75 балів, то, за згодою здобувача освіти, набрані бали можуть бути зараховані як підсумкова оцінка з ОК.

Іспит складається у випадку якщо здобувач освіти бажає підвищити набрану кількість балів із цього ОК. У цьому випадку бали, набрані здобувачем за результатами модульних контрольних робіт (якщо вони передбачені силабусом ОК) анулюються.

Іспит передбачає виконання тестових завдань та розв'язування задач (розробка програми). Викладач залишає за собою право ставити уточнюючі питання під час відповіді студента та просити прокоментувати програмний код. Студент повинен використати засоби тієї мови програмування, яка вказана у завданні.

Передбачається виконання індивідуальних завдань. Варіант ІНДЗ включає себе набір задач, що охоплюють одну або кілька близьких тем. Або одне завдання, розв'язання якого вимагає самостійного опрацювання невеликих тем.

Приклад ІНДЗ (Семестр 2):

Створити клас `СТУДЕНТ`, із зазначенням прізвища, ім'я, по батькові, курсу, ідентифікаційного номера та рейтингу за 100-бальною шкалою.

Визначити конструктори ініціалізації, копіювання, деструктор та методи для зміни і читання значень полів даних класу.

Перевантажити операцію `++` (дружня до класу функція) для зміни курсу, операцію `^` (метод класу) - для зміни рейтингу, операцію присвоєння об'єктів `=`, потокові операції введення `>>` та виведення `<<` об'єктів.

Створити похідний клас - `СТУДЕНТ-ДИПЛОМНИК`, що має тему дипломної роботи (рядок символів) та відсоток її виконання на поточний момент часу. Визначити конструктор, деструктор та метод зміни назви дипломної роботи.

У межах ієрархії класів побудувати поліморфічний кластер на основі віртуального методу для читання полів-рядків символів (імені студента або назви дипломної роботи). Продемонструвати механізм пізнього зв'язування.

Розробити клас `ГРУПА`, який містить масив об'єктів класу `СТУДЕНТ-ДИПЛОМНИК`. Визначити прізвища студентів, відсоток виконання роботи яких становить понад 80 відсотків.

Для роботи з масивом об'єктів побудувати та використати клас-ітератор.

Семестр 1

Питання для ліквідації заліку

1. Мова програмування: поняття, складові частини.
2. Класифікація мов програмування.
3. Принципи структурного програмування. Основні алгоритмічні структури.
4. Компілятори. Системи програмування.
5. Структура програми мовою C++. Основні етапи виконання програми.
6. Імена змінних, константи. Введення-виведення.

7. Арифметичні, логічні оператори. Відношення.
8. Оператори і вирази присвоювання.
9. Управляючі конструкції C++. Умовні вирази.
10. Управляючі конструкції C++. Цикли.
11. Вкладені цикли.
12. Оператори переходу. Приклади.
13. Основні відомості про функції: значення, параметри, аргументи, прототипи функцій.
14. Визначення функцій.
15. Локальні і глобальні змінні.
16. Одновимірні масиви у мові програмування C++. Ініціалізація масивів.
17. Багатовимірні масиви.
18. Рядки. Методи та засоби обробки текстових рядків.
19. Структури та операції з ними. Масиви структур.
19. Потокове введення/виведення у мові програмування C++.
20. Файлове введення/виведення у мові програмування C++.
21. Стандартна бібліотека STL.
22. Механізм використання покажчиків на структури і оператора "стрілка".
23. Поняття про виняткову ситуацію. Обробка виняткових ситуацій.

Семестр 2

Питання для іспиту

1. Поняття об'єкту та його порівняння зі структурами даних та алгоритмів в мові C++.
2. Основні поняття ООП. Класи та об'єкти.
3. Клас та екземпляр. Найпростіший синтаксис означення класу.
4. Специфікатори доступу public, protected та private.
5. Застосування абстрактних класів у мовах C++ та Java.
6. Поняття інкапсуляції. Поняття про захист внутрішніх даних об'єкту у мовах C++ та Java.
7. Конструктори, деструктори, їх роль та призначення.
8. Наслідування. Механізм наслідування членів-даних та методів у мовах C++ та Java.
9. Доступ до членів класу та наслідування у мовах C++ та Java.
10. Відношення клас-підклас та його зв'язок з відношенням абстрактне- конкретне.
Сумісність типів знизу вгору.
11. Особливості виклику конструктора базового класу з конструктору надкласу. Порядок виклику конструкторів та деструкторів для об'єктів похідних класів.
12. Ієрархія класів. Множинне наслідування та його проблеми.
13. Поліморфізм та віртуальні функції. Поняття оголошеного та фактичного типу
14. Механізм виклику віртуальної функції.
15. Дружні функції, окремі класи та класи в цілому, обхід механізмів захисту членів класу.
16. Переваги та недоліки використання механізму дружності в програмах з об'єктно-орієнтованою композицією.
17. Перевантаження операторів функціями та методами.
18. Віртуальні методи у мовах C++ та Java.
19. Шаблони функцій та класів у мовах C++ та Java.
20. Організація захищеного доступу у мовах C++ та Java.

Приклад завдання для іспиту:

1. Засобами мови програмування Java побудувати клас ЧАС, що містить години, хвилини, секунди. Визначити необхідні конструктори, set-, get-методи, деструктор. Перевантажити операцію = для присвоювання часу, операції +=, -= для додавання та віднімання години, відповідно, операції ==, != для порівняння заданих моментів часу.
2. Засобами мови програмування C++ створити клас ОСОБА, що має ім'я, вік, вагу. Визначити конструктори, деструктор і функцію виведення. Створити public-похідний клас ШКОЛЯР, який має рік навчання. Визначити конструктори за замовчуванням та з параметрами, деструктори, функцію виведення. Визначити функції перепризначення віку і класу.

Семестр 3

Питання до 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. Вбудовані модулі та функції для програмування задач, пов'язаних з кібербезпекою.
12. Вбудовані засоби для пошуку та сортування даних в мові Python.
13. Оголошення класу, реалізація класу мовою Python. Конструктор, деструктор.
14. Реалізація інкапсуляції мовою Python.
15. Сетери, гетери, делетери. Статичні поля та методи. Графічна бібліотека turtle та її застосування.
16. Реалізація класу на базі уже створеного класу: композиція.
17. Реалізація класу на базі уже створеного класу: наслідування, множинне наслідування.
18. Реалізація класу на базі уже створеного класу: багаторівневе наслідування. Порядок виклику методів (MRO) в Python.
19. Реалізація поліморфізму мовою Python. Віртуальний метод.
20. Зв'язки між класами: асоціація, агрегація, композиція, спадкування, залежність, реалізація.
21. Спеціальні поля та методи. Особливості перевантаження операторів.
22. Абстрактні класи в мові Python.
23. Поняття про метапрограмування. Декоратори у мові Python.
24. Ітератори та генератори.
25. Інтерфейси у мові Python.
26. Принципи узагальненого програмування в Python.
27. Рефлексія та інтроспекція у мові Python.

28. Бібліотеки Python для роботи з БД.
29. Створення графічних програм GUI за допомогою модуля Tkinter на мові програмування Python. Перша програма. Віджет Button (кнопка).
30. Позиціонування: Pack, Place, Grid. Обробка подій.
31. Особливості, властивості та робота з віджетами: Label, Frame, Text, Scrollbar.
32. Особливості, властивості та робота з віджетами: Entry, Radiobutton, Checkbutton.
33. Особливості, властивості та робота з віджетами: Scale, Listbox, Canvas.
34. Особливості, властивості та робота з віджетами: Menu, MessageBox, звичайні вікна та діалогові.
35. Стилiзація графічних програм GUI за допомогою модуля Tkinter.
36. Дизайн графічного інтерфейсу користувача: бібліотека PyQt. ПЗ QtDesigner.

Приклад завдання до 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.

VI. 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

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

Оцінка в балах за всі види навчальної діяльності	Лінгвістична оцінка
90 – 100	Зараховано
82 – 89	
75 – 81	
67 -74	
60 – 66	
1 – 59	Незараховано (необхідне перескладання)

VI. Рекомендована література та інтернет-ресурси.

Семестр 1

- 1 Гришанович Т. О. Програмування 1 [Електронний ресурс] : електронний курс навчальної дисципліни, затверджений НМР ВНУ імені Лесі Українки, протокол № 4 від 16.12.2020. ВНУ ім. Лесі Українки, 2020. URL: <http://cs.vnu.edu.ua/moodle/course/view.php?id=124>
- 2 Іванов Є.О., Ліндер Я.М., Жереб К.А. Основи мови програмування C++: навчальний посібник. К.: Логос, 2020. 90 с.
- 3 Зеленський О.С., Лисенко В.С. Основи програмування на C++: навчальний посібник. Кривий Ріг: Державний університет економіки і технологій, 2023. 269 с
- 4 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>
- 5 Трофименко О. Г., Прокоп Ю. В., Логінова Н. І., Задерейко О. В. C++. Алгоритмізація та програмування: підручник. Одеса : Фенікс, 2019. 477
- 6 Трофименко О. Г., Прокоп Ю. В., Задерейко О. В. Алгоритмізація та програмування : навчально-методичний посібник. Одеса : Фенікс, 2020. 310 с.
- 7 Hello World - GitHub Docs. *GitHub Docs*. URL: <https://docs.github.com/en/get-started/quickstart/hello-world> (date of access: 20.08.2024).

Семестр 2

- 1 Гришанович Т. О. Програмування 2 [Електронний ресурс] : електронний курс навчальної дисципліни, затверджений НМР ВНУ імені Лесі Українки, протокол № 4 від 16.12.2020. ВНУ ім. Лесі Українки, 2020. URL: <http://cs.vnu.edu.ua/moodle/course/view.php?id=99>.
- 2 Гришанович Т. О., Глинчук Л. Я. Основи об'єктно-орієнтованого програмування : навч. Посібник. ВНУ імені Лесі Українки. Електронні текстові дані (1 файл: 998 КБ). Луцьк : ВНУ імені Лесі Українки, 2022. 120 с. URI: <https://evnuir.vnu.edu.ua/handle/123456789/20320>
- 3 Основи програмування на C++ [Електронний ресурс] : навч. посібник / О. О. Водка [та ін.] ; Нац. техн. ун-т "Харків. політехн. ін-т". Електрон. текст. дані. Харків, 2021. 112

с. –URI: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/52280>.

4. Кадомський К.К., Ніколюк П.К. Java. Теорія і практика: навчальний посібник для студентів природничих спеціальностей університетів. Вінниця: Донну, 2019. 197 с.
5. Підручник JAVA для початківців: 100+ практичних відео-підручників Java - Інший. *Огляди, Ігри, Розваги, Вересень 2022*. URL: <https://uk.myservername.com/java-tutorial-beginners>.
6. Тарнавський Ю. Java-програмування комп'ютерний практикум. Київ : КПІ ім. Ігоря Сікорського, 2021. 90 с.
7. Java Посібник для початківців: Вивчіть ядро Java Програмування. *Guru99*. URL: <https://www.guru99.com/uk/java-tutorial.html> (дата звернення: 12.09.2024).

Семестр 3

1. Глинчук Л.Я. Програмування 3 [Електронний ресурс]: електронний курс навчальної дисципліни, URL: <https://moodle-cs.vnu.edu.ua/course/view.php?id=250>
2. 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.lu/wslgk>
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-Світазь, 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-Світазь, 3-5 червня 2023 р.). Луцьк: Волинський національний університет імені Лесі Українки, 2023. С.67-69.
6. Юрченко І. В., Сікора В. С. Програмування мовою Python: навчальний посібник. Чернівці : Чернів. нац. ун-т. 2022. 104 с.
7. Татарчук Д. Д., Діденко Ю. В. Об'єктно-орієнтоване програмування мовою Python конспект лекцій. Київ : КПІ ім. Ігоря Сікорського, 2021. 128 с.
8. Костюченко А. О. Основи програмування мовою Python навчальний посібник. Чернівці : ФОП Баликіна С.М, 2020. 176 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. Процедурне програмування VS ООП | FoxmindED. *FoxmindEd*. URL: <https://foxminded.ua/protsedurne-prohramuvannia/>
11. Уроки Python - aCode. *aCode*. URL: <https://acode.com.ua/lessons-python/>
12. Python | BestProg. *BestProg* | Програмування: теорія та практика. URL: https://www.bestprog.net/uk/sitemap_ua/python-ua/
13. Python – просто!. *Google Sites: Sign-in*. URL: <https://sites.google.com/comp-sc.if.ua/python-easy/домашня-сторінка>

14. Python – Tkinter. Інформатика – це просто. Bevor Sie zu YouTube weitergehen. URL: <https://www.youtube.com/playlist?list=PLYRayxBfqmqFWkGHLsbg-cARvYdwH3nkoh>
15. Курс: Програмування на Python, Секція: Py Qt 5 | Інформатика - ДистОсвіта. *На головну | Інформатика - ДистОсвіта*. URL: <https://dystosvita.org.ua/course/section.php?id=68>.
16. Bevor Sie zu YouTube weitergehen. *Основи програмування у Python*. URL: https://www.youtube.com/playlist?list=PLqrtuwUbDyeliqb9dT_CzX9nHulMveCLp.
17. Бібліотека Tkinter мови Python. URL: <https://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/tkinter/index.html#Text>