

**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 125 Кібербезпека та захист інформації**  
**освітньо-професійної програми**  
**Кібербезпека та захист інформації**

**Силабус освітнього компонента «Програмування»** підготовки бакалавра, галузі знань 12 Інформаційні технології, спеціальності 125 Кібербезпека та захист інформації, за освітньою програмою Кібербезпека та захист інформації.

**Розробники:**

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

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

Жигаревич О.К., старший викладач кафедри комп'ютерних наук та кібербезпеки

**Погоджено**

Гарант освітньо-професійної



програми:

Глинчук Л.Я.

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

протокол № 2 від 28 вересня 2024 р.

Завідувач кафедри:



Гришанович Т. О.

© Глинчук Л. Я., 2024 р.  
© Гришанович Т. О., 2024 р.  
© Жигаревич О.К., 2024 р.

## I. Опис освітнього компонента

Найменування показників	12 Інформаційні технології 125 Кібербезпека та захист інформації Кібербезпека та захист інформації бакалавр	Характеристика освітнього компонента
Денна форма навчання		Нормативна
Кількість годин/кредитів 570 / 19		Рік навчання 1
		Семестр 1-2-ий
ІНДЗ: є		Лекції 88 год.
		Лабораторні 110 год.
		Самостійна робота 168 год.
	Консультації 24 год.	
	Форма контролю: екзамен, екзамен	
Мова навчання: українська		

## II. Інформація про викладача ПП

Глинчук Людмила Ярославівна

Науковий ступінь кандидат фізико-математичних наук

Посада доцент кафедри комп'ютерних наук та кібербезпеки

Контактна інформація [hlynchuk.ludmila@vnu.edu.ua](mailto:hlynchuk.ludmila@vnu.edu.ua)

Дні занять <http://194.44.187.20/>

ПП Гришанович Тетяна Олександрівна

Науковий ступінь кандидат фізико-математичних наук

Посада доцент кафедри комп'ютерних наук та кібербезпеки

Контактна інформація [hryshanovych.tatiana@vnu.edu.ua](mailto:hryshanovych.tatiana@vnu.edu.ua)

Дні занять <http://194.44.187.20/>

Жигаревич Оксана Костянтинівна

Науковий ступінь

Посада старший викладач кафедри комп'ютерних наук та кібербезпеки

Контактна інформація [Zhyharevych.oksana@vnu.edu.ua](mailto:Zhyharevych.oksana@vnu.edu.ua)

Дні занять <http://194.44.187.20/>

## III. Опис освітнього компонента

- 1. Анотація курсу.** Освітній компонент “Програмування” відноситься до переліку дисциплін циклу професійної підготовки. Навчальна дисципліна присвячена сучасним технологіям програмування: структурного, об’єктно-орієнтованого, логічного, функціонального. 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 паттернами

проектування, із сервісом для спільної розробки програмного забезпечення GitHub та розробкою найпростіших unit-тестів.

**2. 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 комп’ютерних наук, 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, логічного та функціонального програмування; про вбудовані типи даних у мовах програмування C++, Java, 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, застосовувати на практиці паттерни проектування, розробляти unit-тести.

### **3. Результати навчання.**

#### **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.

**ЗК4.** Здатність спілкуватися державною мовою як усно, так і письмово.

**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.

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

**СК3.** Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв’язності та нерозв’язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

**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.

**СК12.** Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.

**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.

#### 4. Структура освітнього компонента.

##### Семестр 1

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/ Бали
<b>Змістовий модуль 1. Вступ до програмування. Мова C++.</b>						
Тема 1. Основні парадигми програмування. Класифікація мов програмування. Структура мови програмування.	6	2		4		ДС
Тема 2. Вступ до мови C++. Структура програми, типи даних.	8	2	2	4		Звіт по лаб. роботі/2
Тема 3. Реалізація базових алгоритмічних конструкцій мовою C++.	14	4	6	4		Звіт по лаб. роботі/4
Тема 4. Функції. Організація рекурсії.	14	4	4	4	2	Звіт по лаб. роботі/4
Тема 6. Адресація в мові C++. Показники.	10	2	2	4	2	Звіт по лаб. роботі/2
Тема 7. Директиви препроцесора.	8	2	2	4		Звіт по лаб. роботі/2
Тема 8. Обробка виняткових ситуацій.	6	2	2	2		Звіт по лаб. роботі/2
Разом за модулем 1	66	18	18	26	4	<b>16</b>
<b>Змістовий модуль 2. Складені типи в мові програмування C++</b>						
Тема 1. Масиви. Особливості реалізації програм із використанням масивів.	18	6	6	6		Звіт по лаб. роботі/6
Тема 2. Файли. Робота з файлами в C++.	12	2	4	6		Звіт по лаб. роботі/4
Тема 3. Система динамічного розподілу пам'яті.	12	2	2	6	2	Звіт по лаб. роботі/2

Тема 4. Рядки. Особливості реалізації програм із використанням текстових рядків.	12	4	4	4		Звіт по лаб. роботі/4
Тема 5. Структури. Особливості реалізації програм із використанням структур та масивів структур.	12	4	4	4		Звіт по лаб. роботі/4
Тема 6. Об'єднання та переліки. Особливості реалізації програм із використанням об'єднань та переліків.	8	2	2	2	2	Звіт по лаб. роботі/2
Тема 7. Стандартна бібліотека шаблонів (STL).	10	2	2	4	2	Звіт по лаб. роботі/2
Разом за модулем 2	84	22	24	34	6	<b>24</b>
Тестування						20
Модульна контрольна робота 1						10
Модульна контрольна робота 2						10
ІНДЗ 1						10
ІНДЗ 2						10
<b>Всього годин/Балів</b>	<b>150</b>	<b>40</b>	<b>42</b>	<b>58</b>	<b>10</b>	<b>100</b>

## Семестр 2

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

Тема 9. Абстрактні класи.	10	2	2	4	2	Звіт по лаб. роботі/2
Разом за модулем 3	74	20	20	30	4	<b>20</b>
<b>Змістовий модуль 4. Об'єктно-орієнтоване програмування мовою Java</b>						
Тема 1. Вступ до мови програмування Java. Основні поняття. Базовий синтаксис, структура програми.	8	2	2	2	2	Звіт по лаб. роботі/2
Тема 2. Типи даних в мові Java, реалізація базових алгоритмічних структур.	6	2	2	2		Звіт по лаб. роботі/2
Тема 3. Дані та методи класу. Досяжність. Реалізація інкапсуляції у мові Java.	12	4	4	4		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.	8	2	2	4		Звіт по лаб. роботі/2
Тема 6. Реалізація поліморфізму в мові Java.	10	4	2	2	2	Звіт по лаб. роботі/2
Тема 7. Створення unit-тестів в середовищі IntelliJ IDEA.	12	2	4	6		Звіт по лаб. роботі/6
Разом за модулем 4	62	18	18	22	4	<b>20</b>
Тестування						20
Модульна контрольна робота 1						10
Модульна контрольна робота 2						10
ІНДЗ 1						10
ІНДЗ 2						10
<b>Всього годин/Балів</b>	<b>136</b>	<b>38</b>	<b>38</b>	<b>52</b>	<b>8</b>	<b>100</b>

### Семестр 3

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/Бали
<b>Змістовий модуль 5. Вступ до програмування мовою Python</b>						
Тема 1. Вступ до мови програмування Python. Особливості мови. Структура програми.	6	2	2	2		Звіт по лаб. роботі/1
Тема 2. Вбудовані типи даних. Змінні та літерали.	8	2	2	4		Звіт по лаб. роботі/2
Тема 3. Винятки та їх обробка.	8	2	2	4		Звіт по лаб. роботі/1
Тема 4. Реалізація базових алгоритмічних структур мовою Python.	10	2	2	4		Звіт по лаб. роботі/2

Тема 5. Структури даних в Python: списки, кортежі, словники, рядкові величини, множини.	10	2	4	4		Звіт по лаб. роботі/4
Тема 6. Функції, лямбда функції та рекурсія.	10	2	2	4	2	Звіт по лаб. роботі/2
Тема 7. Робота з файлами в Python.	8	2	2	4		Звіт по лаб. роботі/2
Тема 8. Модулі в Python.	10	2	2	4	2	Звіт по лаб. роботі/2
Тема 9. Створення unit-тестів для мови Python	4	2	4			Звіт по лаб. роботі/2
Разом за модулем 5	74	18	22	30	4	<b>18</b>
<b>Змістовий модуль 6. Об'єктно-орієнтоване програмування мовою Python</b>						
Тема 1. Оголошення класу, реалізація класу мовою Python. Конструктор, деструктор.	8	2	2	4		Звіт по лаб. роботі/1
Тема 2. Реалізація інкапсуляції мовою Python. Сетери, гетери, делетери. Статичні поля та методи.	8	2	2	4		Звіт по лаб. роботі/2
Тема 3. Реалізація класу на базі уже створеного класу: композиція.	8	2	2	2	2	Звіт по лаб. роботі/1
Тема 4. Реалізація класу на базі уже створеного класу: наслідування, множинне наслідування, багаторівневе наслідування. Порядок виклику методів (MRO) в Python	8	2	4	2		Звіт по лаб. роботі/4
Тема 5. Спеціальні поля та методи. Особливості перевантаження операторів.	8	2	2	4		Звіт по лаб. роботі/2
Тема 6. Поняття про метапрограмування. Декоратори.	10	2	2	4	2	Звіт по лаб. роботі/2
Тема 7. Реалізація поліморфізму мовою Python.	10	2	4	4		Звіт по лаб. роботі/2
Разом за модулем 6	60	14	18	24	4	<b>14</b>
<b>Змістовий модуль 7. Паттерни проєктування</b>						
Тема 1. Паттерни проєктування: породжуючі паттерни.	6	2	2	2		Звіт по лаб. роботі/2
Тема 3. Паттерни проєктування: структурні паттерни.	6	2	2	2		Звіт по лаб. роботі/2
Тема 3. Паттерни проєктування: поведінкові паттерни.	18	4	6	6	2	Звіт по лаб. роботі/4
Разом за модулем 7	30	8	10	10	2	<b>8</b>
Тестування						20
Модульна контрольна робота 1						10
Модульна контрольна робота 2						10
ІНДЗ 1						10



ІНДЗ 2						10
<b>Всього годин/Балів</b>	164	40	50	64	10	100

#### Семестр 4

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/ Бали
<b>Змістовий модуль 7. Основні методи та підходи у функціональному програмуванні</b>						
Тема 1. Вступ до функціонального програмування. Мови функціонального програмування.	4	2		2		ДС
Тема 2. Вступ до мови Scala. Найпростіші функції.	8	2	4	2		Звіт по лаб. роботі/4
Тема 3. Анонімні функції. Каррінг.	8	2	2	4		Звіт по лаб. роботі/2
Тема 4. Лінійні обчислення у функціональних мовах програмування. Рекурсія.	10	2	4	4		Звіт по лаб. роботі/4
Тема 5. Функції вищих порядків.	12	2	4	4	2	Звіт по лаб. роботі/4
Тема 6. Співставлення із зразком.	10	2	2	4	2	Звіт по лаб. роботі/2
Тема 7. Класи та об'єкти у мові Scala.	10	2	4	4		Звіт по лаб. роботі/4
Разом за модулем 1	62	14	20	24	4	<b>20</b>
<b>Змістовий модуль 8. Основні методи та підходи у логічному програмуванні</b>						
Тема 1. Вступ до логічного програмування. Мови логічного програмування.	6	2		4		ДС
Тема 2. Вступ до мови Prolog. Синтаксичні особливості та вбудовані предикати.	8	2	2	4		Звіт по лаб. роботі/2
Тема 3. Рекурсивне представлення даних і програм.	10	2	2	4		Звіт по лаб. роботі/2
Тема 4. Метод відсікання та метод повернення в логічному програмуванні.	10	2	4	4		Звіт по лаб. роботі/4
Тема 5. Методи організації повторень та рекурсії мовою Prolog.	10	2	4	4		Звіт по лаб. роботі/4
Тема 6. Побудова прикладних програм мовою логіки: створення бази даних.	6	2	4	2		Звіт по лаб. роботі/4
Тема 7. Побудова прикладних програм мовою логіки: побудова експертних систем.	6	2	2	2	2	Звіт по лаб. роботі/4
Разом за модулем 2	58	14	18	24	2	<b>20</b>
<b>Види підсумкових робіт</b>						Бал

Тестування						20
Модульна контрольна робота 1						10
Модульна контрольна робота 2						10
ІНДЗ 1						10
ІНДЗ 2						10
<b>Всього годин/Балів</b>	120	28	38	48	6	100
<b>Всього годин за курс</b>	570	146	168	228	28	

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

№ з/п	Тема	Кількість годин
1	Типи даних у С++ для представлення часу та дати.	4
2	Макроси у мові С++.	4
3	Організація форматowanego виводу. Бібліотека <cstdio>.	2
4	Порозрядні оператори (I, АБО, НЕ).	4
5	Специфікатори типів даних.	2
6	Механізм використання покажчиків на структури і оператора "стрілка".	4
7	Поняття про бітові поля структур.	2
8	Механізм використання об'єднань. Анонімні об'єднання.	4

<b>Всього</b>	26
---------------	----

### **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. 22 год  
Перевірка здійснюється під час модульних контрольних заходів і оцінюється відповідною кількістю балів.

<b>№ з/п</b>	<b>Тема</b>	<b>Кількість годин</b>
<b>1</b>	Ідентифікація та приведення типів в C++.	2
<b>2</b>	Види класів в C++.	6
<b>3</b>	Класи потокового введення-виведення.	4
<b>4</b>	Види класів в мові Java.	4
<b>5</b>	Абстрактні класи та абстрактні методи в Java.	4
<b>6</b>	Клас Math у Java.	2
	<b>Всього</b>	<b>22</b>

### **Семестр 3**

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. 22 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. 24 год  
 Перевірка здійснюється під час модульних контрольних заходів і оцінюється відповідною кількістю балів.

№ з/п	Тема	Кількість годин
1	Визначення функції із списком аргументів у Python.	4
2	Вбудовані модулі. Дата та час у Python. Форматування дат (Date Formatting).	4
3	Вбудовані засоби для пошуку та сортування даних в мові Python.	4
4	Абстрактні класи в мові Python.	4
5	Метакласи у мові Python.	4
6	Templates в Python.	4
7	Організація unit-тестування у середовищах розробки.	4
	<b>Всього</b>	<b>28</b>

#### Семестр 4

Опрацювання лекційного матеріалу. 6 год  
 Перевірка здійснюється під час лабораторних занять та оцінюється при виставленні оцінки за змістовий модуль.

Підготовка до лабораторних занять, виконання домашніх завдань. 10 год  
 Перевірка здійснюється під час лабораторних занять.

Систематизація вивченого матеріалу перед іспитом. 4 год  
 Перевірка здійснюється під час іспиту.

Вивчення тем, що не розглядаються в курсі лекцій. 28 год  
 Перевірка здійснюється під час модульних контрольних заходів і оцінюється відповідною кількістю балів.

№ з/п	Тема	Кількість годин
1	Мова програмування Haskell. Особливості.	4
2	Аналіз великих даних у Scala.	4
3	Мова Scala та паралельні обчислення.	4
4	Моноїди та монади. Функтори.	2
5	Операції над множинами в мові Prolog.	2
6	Типізація та Каррі. Типізація за Чьорчем.	4
7	Робота з динамічною базою даних.	4
8	Мова програмування DataLog. Особливості та сфери застосування.	4
	<b>Всього</b>	<b>28</b>

#### **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, Програмування 2, Програмування 3, Програмування 4, розміщених на платформах дистанційного навчання Moodle. (<https://moodlecs.vnu.edu.ua/course/view.php?id=74>, <https://moodle-cs.vnu.edu.ua/course/view.php?id=85>, <https://moodle.vnu.edu.ua/course/view.php?id=886>). 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.

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 не проводиться.

У третьому семестрі, якщо здобувач самостійно набув компетентностей із програмування мовою Python, використовуючи дистанційний курс “Основи програмування” (<https://prometheus.org.ua/prometheus-plus/python-beetroot-course/>) та отримав сертифікат про завершення курсу, йому буде зараховано Теми 1, 2, 4, 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 зараховано змістовий модуль 5 із результатом 16 балів.

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».

#### **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 перекладання**

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”, “Програмування 4”, 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. Перекладання контрольних робіт та тестувань заборонено.

### **V. Підсумковий контроль**

Підсумковою формою контролю освітнього компонента “Програмування” є іспит у 2 та 4 семестрах та залік у 1 та 3 семестрах.

Для 1 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 за семестр – 100 балів. Залік виставляється за результатами поточної роботи за умови, що здобувач освіти виконав усі види навчальної діяльності, визначеної силабусом освітнього компонента.

У випадку, якщо здобувач освіти не відвідував окремі аудиторні заняття (з поважних причин), на консультаціях він має право відпрацювати пропущені заняття та здати завдання. У дату складання заліку викладач записує у відомість суму поточних балів, які здобувач освіти набрав під час семестру.

Якщо протягом семестру здобувач освіти набрав менше, ніж 60 балів, він складає залік під час ліквідації академічної заборгованості. У цьому випадку бали, набрані під час поточного оцінювання анулюються. Максимальна кількість балів, яку може отримати здобувач під час ліквідації академічної заборгованості, становить 100. 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 4 семестрів підсумковою формою контролю освітнього компонента є іспит. Оцінювання навчальних досягнень здійснюється за 100 бальною шкалою. Оцінка включає в себе поточний контроль (оцінюється робота на парях, 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 – 40 балів. Максимальна кількість балів, яку може отримати здобувач за підсумковий контроль за семестр складає 60 балів.

Якщо за результатами семестру накопичено не менше 75 балів і студент погоджується із цим результатом, то оцінка за семестр може виставлятися без складання іспиту. В іншому випадку студент складає іспит; максимальна кількість балів, яку можна отримати на іспиті – 60 балів. Вони замінюють бали модульного семестрового контролю, поточний семестровий контроль при цьому зберігається.

Іспит передбачає виконання тестових завдань (30 питань по одному балу) та розв'язування задач (розробка програми) – 2 задачі по 15 балів. Викладач залишає за собою право ставити уточнюючі питання під час відовіді студента та просити прокоментувати програмний код. Студент повинен використати засобі 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. Потокowe 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. Механі
9. зм наслідування членів-даних та методів у мовах C++ та Java.
10. Доступ до членів класу та наслідування у мовах C++ та Java.
11. Відношення клас-підклас та його зв'язок з відношенням абстрактне- конкретне. Сумісність типів знизу вгору.
12. Особливості виклику конструктора базового класу з конструктору надкласу. Порядок виклику конструкторів та деструкторів для об'єктів похідних класів.
13. Ієрархія класів. Множинне наслідування та його проблеми.
14. Поліморфізм та віртуальні функції. Поняття оголошеного та фактичного типу
15. Механізм виклику віртуальної функції.
16. Дружні функції, окремі класи та класи в цілому, обхід механізмів захисту членів класу.
17. Переваги та недоліки використання механізму дружності в програмах з об'єктно-орієнтованою композицією.
18. Перевантаження операторів функціями та методами.
19. Віртуальні методи у мовах C++ та Java.
20. Шаблони функцій та класів у мовах C++ та Java.
21. 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 для ліквідації екзамену

1. Вступ до мови програмування Python. Особливості мови. Структура програми.
2. Вбудовані типи даних. Змінні та літерали.
3. Винятки та їх обробка.
4. Реалізація базових алгоритмічних структур мовою Python.
5. Структури даних у Python: списки, кортежі.
6. Структури даних у Python: словники, рядкові величини, множини.
7. Функції, лямбда функції та рекурсія.
8. Робота з файлами в Python.
9. Модулі в Python.
10. Створення unit-тестів для мови Python.
11. Оголошення класу, реалізація класу мовою Python. Конструктор, деструктор.
12. Реалізація інкапсуляції мовою Python. Сетери, гетери, делетери. Статичні поля та методи.
13. Реалізація класу на базі уже створеного класу: композиція.
14. Реалізація класу на базі уже створеного класу: наслідування, множинне наслідування.
15. Реалізація класу на базі уже створеного класу: багаторівневе наслідування. Порядок виклику методів (MRO) в Python.
16. Спеціальні поля та методи. Особливості перевантаження операторів.
17. Поняття про метапрограмування. Декоратори у мові Python.
18. Реалізація поліморфізму мовою Python.
19. Абстрактні класи в мові Python.
20. Метакласи у мові Python.
21. Templates (шаблони) в Python.
22. Вбудовані та зовнішні модулі. Робота з датою та часом.
23. Паттерни проєктування: породжуючі паттерни.
24. Паттерни проєктування: структурні паттерни.
25. Паттерни проєктування: поведінкові паттерни.

### Семестр 4

#### Питання для іспиту

1. Вступ до функціонального програмування. Мови функціонального програмування.
2. Вступ до мови Scala, її особливості.

3. Реалізація найпростіших функцій. Приклади.
4. Анонімні функції. Приклади.
5. Каррінг. Приклади.
6. Лінійні обчислення у функціональних мовах програмування.
7. Рекурсія у мові Scala.
8. Функції вищих порядків.
9. Співставлення із зразком.
10. Класи та об'єкти у мові Scala.
11. Аналіз великих даних у Scala.
12. Мова Scala та паралельні обчислення.
13. Моноїди та монади. Функтори.
14. Особливості мови Haskell.
15. Вступ до логічного програмування. Мови логічного програмування.
16. Вступ до мови Prolog. Синтаксичні особливості та вбудовані предикати.
17. Мова Prolog: рекурсивне представлення даних і програм.
18. Мова Prolog: метод відсікання та метод повернення в логічному програмуванні.
19. Методи організації повторень та рекурсії мовою Prolog.
20. Побудова прикладних програм мовою логіки: створення бази даних.
21. Побудова прикладних програм мовою логіки: побудова експертних систем.
22. Операції над множинами в мові Prolog.
23. Типізація та Каррі. Типізація за Чьорчем.
24. Робота з динамічною базою даних.
25. Мова програмування DataLog. Особливості та сфери застосування.

Приклад завдання для іспиту:

1. Засобами мови Scala реалізувати рекурсивну функцію перестановки місцями сусідніх парних і непарних елементів в заданому списку.
2. Засобами мови Prolog реалізувати програму, яка повертатиме псевдовипадкові дійсні числа з діапазону (0; 1).

## 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. 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>
4. Трофименко О. Г., Прокоп Ю. В., Логінова Н. І., Задерейко О. В. C++. Алгоритмізація та програмування: підручник. Одеса : Фенікс, 2019. 477
5. Трофименко О. Г., Прокоп Ю. В., Задерейко О. В. Алгоритмізація та програмування : навчально-методичний посібник. Одеса : Фенікс, 2020. 310 с.

**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. Кадомський К.К., Ніколюк П.К. Java. Теорія і практика: навчальний посібник для студентів природничих спеціальностей університетів. Вінниця: Донну, 2019. 197 с.
4. Підручник JAVA для початківців: 100+ практичних відео-підручників Java - Інший. *Огляди, Ігри, Розваги, Вересень 2022*. URL: <https://uk.myservername.com/java-tutorialbeginners>.
5. Тарнавський Ю. Java-програмування комп'ютерний практикум. Київ : КПІ ім. Ігоря Сікорського, 2021. 90 с.
6. Hello World - GitHub Docs. *GitHub Docs*. URL: <https://docs.github.com/en/get-started/quickstart/hello-world> (date of access: 28.08.2023).

### Семестр 3

1. Глинчук Л.Я. Програмування 3 [Електронний ресурс]: електронний курс навчальної дисципліни, URL: <https://moodle.vnu.edu.ua/course/view.php?id=886>
2. Костюченко А. О. Основи програмування мовою Python навчальний посібник. Чернівці : ФОП Баликіна С.М, 2020. 176 с.
3. Татарчук Д. Д., Діденко Ю. В. Об'єктно-орієнтоване програмування мовою Python конспект лекцій. Київ : КПІ ім. Ігоря Сікорського, 199. 128 с.
4. Юрченко І. В., Сікора В. С. Програмування мовою Python: навчальний посібник. Чернівці : Чернів. нац. ун-т. 2022. 104 с.
5. Яковенко А. В. Основи програмування. Python. Частина 1. підручник для студ. спеціальності 122 "Комп'ютерні науки". Київ : КПІ ім. Ігоря Сікорського, 2018. Ч. 1. 195 с.
6. Фрімен Е. Head First. Патерни проектування. Київ : Фабула, 2020. 672 с.

### Семестр 4

1. Заяць В. М. Логічне і функціональне програмування. Системний підхід : підручник / В. М. Заяць, М. М. Заяць. Рівне : НУВГП, 2018. 422 с
2. Шевченко І.В. Функціональне та логічне програмування (Частина 1. Функціональне програмування): Навч. посібник з виконання лабораторних робіт. Харків: Нац. аерокосм. ун-т «Харк.авіац. ін-т », 2020. 92с.
3. Документація Scala. URL: <https://docs.scala-lang.org/uk//>