

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Волинський національний університет імені Лесі Українки**  
**Факультет інформаційних технологій і математики**  
**Кафедра загальної математики та методики навчання інформатики**

**СИЛАБУС**  
**нормативного освітнього компонента**

**ПРОГРАМУВАННЯ**

**Підготовки бакалавра**  
**Предметної спеціальності 014.09 Середня освіта (Інформатика)**  
**освітньо-професійної програми Середня освіта. Інформатика**

Луцьк – 2025

**Силабус освітнього компонента «ПРОГРАМУВАННЯ»** підготовки бакалавра, галузі знань 01 Освіта / Педагогіка, предметної спеціальності 014 Середня освіта (Інформатика), за освітньою програмою «Середня освіта. Інформатика»

**Розробник:** Гришанович Тетяна Олександрівна, кандидат фізико-математичних наук  
Пастернак Вікторія Валентинівна, кандидат технічних наук, доцент

**Погоджено**

Гарант освітньо-професійної програми:



Світлана ЯЦЮК

**Силабус освітнього компонента затверджено на засіданні кафедри загальної математики та методики навчання інформатики**  
протокол №2 від 15 вересня 2025 р.

Завідувач кафедри:



Марія ХОМЯК

## I. Опис освітнього компонента

| Найменування показників                   | Галузь знань, спеціальність, предметна спеціальність, освітньо-професійна /освітньо-наукова/освітньо-творча програма, освітній рівень           | Характеристика освітнього компонента                     |
|---|---|--|
| <b>Денна (очна) форма навчання</b>        | 01 Освіта,<br>A4 Середня освіта (Інформатика),<br>A4.09 Середня освіта (Інформатика),<br>Середня освіта (Інформатика)<br>перший (бакалаврський) | <b>Нормативний</b>                                       |
| <b>Кількість годин/кредитів</b><br>480/16 |   | <b>Рік навчання – 1-ий, 2-ий</b>                         |
|   |   | <b>Семестр – 1-ий, 2-ий, 3-ий, 4-ий</b>                  |
|   |   | <b>Лекції – 120 год.</b>                                 |
|   |   | <b>Практичні – 0 год.</b>                                |
|   |   | <b>Лабораторні – 134 год.</b>                            |
| <b>ІНДЗ: немає</b>                        | <b>Індивідуальні – 0 год.</b>   |  |
|   | <b>Самостійна робота – 202 год.</b>   |  |
|   | <b>Консультації – 24 год.</b>   |  |
|   |   | <b>Форма контролю: залік, екзамен<br/>залік, екзамен</b> |
| <b>Мова навчання – українська</b>         |   |  |

## II. Інформація про викладача (- ів)

ППП Пастернак Вікторія Валентинівна  
 Науковий ступінь кандидат технічних наук  
 Вчене звання доцент  
 Посада доцент кафедри загальної математики та методики навчання інформатики  
 Контактна інформація [pasternak.viktorii@vnu.edu.ua](mailto:pasternak.viktorii@vnu.edu.ua)  
 Дні занять <http://194.44.187.20/cgi-bin/timetable.cgi>

## III. Опис освітнього компонента

### 1. Анотація курсу

Освітній компонент «Програмування» є невід'ємною складовою підготовки майбутніх вчителів інформатики та спрямований на підвищення рівня формування у здобувачів освіти знань та умінь, які дадуть теоретичний і практичний фундамент, необхідний для вивчення загальних принципів програмування. Курс охоплює базові концепції програмування, такі як типи даних, алгоритми, цикли, розгалуження, масиви та структури даних, а також об'єктно-орієнтоване програмування. Слід також відмітити, що ОК охоплює навички програмування класичних задач, від розробки алгоритму до його реалізації у вигляді коду, тестування та налагодження на мовах програмування C++, C#, Java, JavaScript або Python. Окрім технічних аспектів, акцент робиться на методиках викладання цих тем учням, а також на формуванні у здобувачів освіти навичок пояснення складних понять у доступній формі.

**2. Пререквізити (попередні курси, на яких базується вивчення освітнього компонента)**

«Алгоритми та структури даних», «Алгоритмізація та програмування», «Дискретна математика».

### 3. Мета і завдання освітнього компонента

Метою освітнього компонента «Програмування» є забезпечити здобувачів освіти методичними та практичними інструментами для викладання основ програмування у закладах загальної середньої освіти.

Завдання освітнього компонента:

- 1) ознайомлення з основними поняттями, принципами та методами програмування;
- 2) вивчення синтаксису та семантики однієї або кількох мов програмування;
- 3) формування навичок побудови алгоритмів та структурованих програм;
- 4) розвинути вміння аналізувати, тестувати та налагоджувати програмний код з метою забезпечення його коректності та оптимальної роботи;
- 5) сформувані практичні навички створення ефективних програмних рішень для розв'язування реальних прикладних задач;
- 6) забезпечити дотримання принципів структурності, читабельності та документування програмного коду.

### 4. Результати навчання (Компетентності)

*Результати навчання:*

РН7. *Застосовувати* систематизовані наукові знання в професійній діяльності відповідно до предметної спеціальності, *оперувати* базовими категоріями та поняттями предметної області спеціальності.

ПРН1. *Визначати* структуру предметної галузі інформатики, її місце в системі наук, *визначати* перспективи розвитку інформатики та інформаційних технологій, їхнє суспільне значення.

ПРН2. *Знати* та *розуміти* фізичні, логічні та математичні основи інформаційних технологій.

ПРН5. *Визначати* та *застосовувати* методи розробки алгоритмів розв'язування задач з інформатики, реалізовувати їх мовами програмування, оцінювати ефективність алгоритмів.

ПРН9. *Розв'язувати* задачі курсу інформатики різного рівня складності, аналізувати та оцінювати ефективність розв'язку та формувати відповідні вміння в учнів.

ПРН13. *Передбачати* та *оцінювати* результати власної діяльності, *аналізувати* перспективний педагогічний досвід з урахуванням закономірностей освітнього процесу закладів загальної середньої освіти.

*Компетентності:*

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу, до застосування знань у практичних ситуаціях.

ЗК2. Знання й розуміння предметної області та професійної діяльності.

ЗК4. Здатність орієнтуватися в інформаційному просторі, здійснювати пошук, аналіз та обробку інформації з різних джерел, ефективно використовувати цифрові ресурси та технології в освітньому процесі.

ЗК8. Здатність зберігати та примножувати моральні, культурні, наукові цінності та досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та значення у розвитку суспільства, техніки і технологій.

ФК1. Здатність застосовувати систематизовані наукові знання в професійній діяльності відповідно до предметної спеціальності.

- ПК1. Здатність використовувати знання наукових фактів, концепцій, теорій, принципів і методів сучасної інформатики у практиці навчання інформатики.
- ПК3. Здатність до використання сучасних методів розробки та дослідження алгоритмів розв'язування задач у моделюванні об'єктів і процесів та реалізації цих алгоритмів сучасними мовами програмування.
- ПК6. Здатність розв'язувати задачі різного рівня складності з курсу інформатики закладів загальної середньої освіти, аналізувати та оцінювати ефективність розв'язку та формувати відповідні вміння в учнів.
- ПК7. Здатність добирати та використовувати сучасні інформаційно-комунікаційні технології в освітньому процесі та в позакласній роботі, аналізувати й оцінювати доцільність й ефективність їх застосування.
- ПК8. Здатність до цифрового подання та обробки текстової, числової, графічної, звукової та відеоінформації.

## 5. 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         |       |                        |
| Тема 2. Вступ до мови C++. Структура програми, типи даних.   | 8      | 2    | 2      | 4         |       | Звіт по лаб. роботі/3  |
| Тема 3. Реалізація базових алгоритмічних конструкцій мовою C++.  | 12     | 2    | 8      | 2         |       | Звіт по лаб. роботі/12 |
| Тема 4. Функції. Організація рекурсії.   | 8      | 2    | 2      | 4         |       | Звіт по лаб. роботі/3  |
| Тема 5. Адресація в мові C++. Показники.   | 10     | 2    | 2      | 4         | 2     | Звіт по лаб. роботі/3  |
| Разом за модулем 1   | 44     | 10   | 14     | 18        | 2     | <b>21</b>              |
| <b>Змістовий модуль 2. Складені типи в мові програмування C++</b>                                      |        |      |        |           |       |                        |
| Тема 1. Масиви. Особливості реалізації програм із використанням масивів.                               | 14     | 2    | 4      | 8         |       | Звіт по лаб. роботі/6  |
| Тема 2. Файли. Робота з файлами в C++.   | 6      | 2    | 2      | 2         |       | Звіт по лаб. роботі/3  |
| Тема 3. Рядки. Особливості реалізації програм із використанням текстових рядків.                       | 6      | 2    | 2      | 2         |       | Звіт по лаб. роботі/3  |
| Тема 4. Структури. Особливості реалізації програм із використанням структур та масивів структур.       | 8      | 2    | 2      | 4         |       | Звіт по лаб. роботі/3  |
| Тема 5. Стандартна бібліотека шаблонів (STL).  | 8      | 2    | 2      | 2         | 2     | Звіт по лаб. роботі/4  |
| Разом за модулем 2   | 42     | 10   | 12     | 18        | 2     | <b>19</b>              |
| <b>Види підсумкових робіт</b>  |        |      |        |           |       | Бал                    |
| Тестування   |        |      |        |           |       | 30                     |
| Модульна контрольна робота 1   |        |      |        |           |       | 15                     |
| Модульна контрольна робота 2   |        |      |        |           |       | 15                     |
| <b>Всього годин/Балів</b>  | 86     | 20   | 26     | 36        | 4     | 100                    |

## Семестр 2

| Назви змістових модулів і тем   | Усього | Лек. | Лабор. | Сам. роб. | Конс. | Форма контролю/<br>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      | 2    | 2      | 4         |       | 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 | 12     | 2    | 4      | 4         | 2     | 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  | 76     | 18   | 20     | 32        | 6     | <b>20</b>               |
| <b>Змістовий модуль 4. Візуальні середовища об'єктно-орієнтованого програмування</b>                |        |      |        |           |       |                         |
| Тема 1. Огляд візуального середовища програмування Scratch.   | 12     | 4    | 4      | 4         |       | Звіт по лаб. роботі/4   |
| Тема 2. Розробка Scratch-проектів, які використовують логічні та арифметичні вирази.                | 10     | 4    | 2      | 4         |       | Звіт по лаб. роботі/2   |
| Тема 3. Розробка Scratch-проектів для малювання фігур.  | 8      | 2    | 2      | 4         |       | Звіт по лаб. роботі/2   |
| Тема 4. Огляд візуального середовища програмування Alice.   | 10     | 2    | 4      | 4         |       | Звіт по лаб. роботі/4   |
| Тема 5. Розробка Alice-проектів, які використовують логічні та арифметичні вирази.                  | 18     | 4    | 6      | 8         |       | Звіт по лаб. роботі/6   |
| Тема 6. Ігрове програмування.   | 12     | 2    | 2      | 6         | 2     | Звіт по лаб. роботі/2   |
| Разом за модулем 4  | 70     | 18   | 20     | 30        | 2     | <b>20</b>               |
| <b>Види підсумкових робіт</b>   |        |      |        |           |       | Бал                     |
| Тестування  |        |      |        |           |       | 40                      |
| Модульна контрольна робота 1  |        |      |        |           |       | 10                      |
| Модульна контрольна робота 2  |        |      |        |           |       | 10                      |
| <b>Всього годин/Балів</b>   | 146    | 36   | 40     | 62        | 8     | 100                     |

### 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.       | 8      | 4    | 2      | 2         |       | Звіт по лаб. роботі/2 |
| Тема 2. Вбудовані типи даних. Змінні та літерали.                                       | 6      | 2    | 2      | 2         |       | Звіт по лаб. роботі/2 |
| Тема 4. Реалізація базових алгоритмічних структур мовою Python.                         | 8      | 2    | 4      | 2         |       | Звіт по лаб. роботі/4 |
| Тема 5. Структури даних в Python: списки, кортежі, словники, рядкові величини, множини. | 8      | 2    | 2      | 4         |       | Звіт по лаб. роботі/2 |
| 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.  | 8      | 2    | 2      | 4         |       | Звіт по лаб. роботі/2 |
| Разом за модулем 5  | 56     | 16   | 16     | 22        | 2     | <b>16</b>             |
| <b>Змістовий модуль 6. Об'єктно-орієнтоване програмування мовою Python</b>              |        |      |        |           |       |                       |
| Тема 1. Оголошення класу, реалізація класу мовою Python.                                | 8      | 2    | 2      | 4         |       | Звіт по лаб. роботі/2 |
| Тема 2. Реалізація інкапсуляції мовою Python. Конструктор, деструктор.                  | 10     | 2    | 4      | 4         |       | Звіт по лаб. роботі/4 |
| Тема 3. Реалізація класу на базі уже створеного класу. Композиція та наслідування.      | 12     | 4    | 4      | 4         |       | Звіт по лаб. роботі/4 |
| Тема 4. Особливості перевантаження операторів.  | 12     | 2    | 4      | 6         |       | Звіт по лаб. роботі/6 |
| Тема 5. Декоратори.   | 12     | 2    | 2      | 6         | 2     | Звіт по лаб. роботі/4 |
| Тема 6. Реалізація поліморфізму мовою Python.   | 10     | 2    | 2      | 4         | 2     | Звіт по лаб. роботі/4 |
| Разом за модулем 6  | 64     | 14   | 18     | 28        | 4     | <b>24</b>             |
| <b>Види підсумкових робіт</b>   |        |      |        |           |       | Бал                   |
| Тестування  |        |      |        |           |       | 20                    |
| Модульна контрольна робота 1  |        |      |        |           |       | 20                    |
| Модульна контрольна робота 2  |        |      |        |           |       | 20                    |
| <b>Всього годин/Балів</b>   |        |      |        |           |       | <b>100</b>            |

### Семестр 4

| Назви змістових модулів і тем  | Усього | Лек. | Лабор. | Сам. роб. | Конс. | Форма контролю/ Бали  |
|--|--------|------|--------|-----------|-------|-----------------------|
| <b>Змістовий модуль 7. Об'єктно-орієнтоване програмування мовою Java</b> |        |      |        |           |       |                       |
| Тема 1. Вступ до мови програмування Java. Основні                        | 10     | 2    | 2      | 6         |       | Звіт по лаб. роботі/2 |

|  |    |    |    |    |   |                       |
|--|----|----|----|----|---|-----------------------|
| поняття. Базовий синтаксис, структура програми.                                      |    |    |    |    |   |                       |
| Тема 2. Типи даних в мові Java, реалізація базових алгоритмічних структур.           | 8  | 2  | 2  | 4  |   | Звіт по лаб. роботі/2 |
| Тема 3. Дані та методи класу. Досяжність. Реалізація інкапсуляції у мові C++.        | 8  | 2  | 2  | 4  |   | Звіт по лаб. роботі/2 |
| Тема 4. Масив об'єктів у Java: Як створити, ініціалізувати та використовувати.       | 8  | 2  | 2  | 4  |   | 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. Створення unit-тестів в середовищі IntelliJ IDEA.                            | 14 | 2  | 2  | 8  | 2 | Звіт по лаб. роботі/4 |
| Разом за модулем 7   | 56 | 12 | 12 | 30 | 2 | <b>14</b>             |
| <b>Змістовий модуль 8. Основні методи та підходи у функціональному програмуванні</b> |    |    |    |    |   |                       |
| Тема 1. Вступ до функціонального програмування. Мови функціонального програмування.  | 6  | 2  | 2  | 2  |   | Звіт по лаб. роботі/2 |
| Тема 2. Вступ до мови Scala. Найпростіші функції.                                    | 6  | 2  | 2  | 2  |   | Звіт по лаб. роботі/2 |
| Тема 3. Анонімні функції.  | 6  | 2  | 2  | 2  |   | Звіт по лаб. роботі/2 |
| Тема 4. Лінійні обчислення у функціональних мовах програмування. Рекурсія.           | 6  | 2  | 2  | 2  |   | Звіт по лаб. роботі/2 |
| Тема 5. Функції вищих порядків.  | 8  | 2  | 2  | 2  | 2 | Звіт по лаб. роботі/2 |
| Тема 6. Класи та об'єкти у мові Scala.   | 6  | 2  | 2  | 2  |   | Звіт по лаб. роботі/2 |
| Разом за модулем 1   | 38 | 12 | 12 | 12 | 2 | <b>12</b>             |
| <b>Змістовий модуль 9. Основні методи та підходи у логічному програмуванні</b>       |    |    |    |    |   |                       |
| Тема 1. Вступ до логічного програмування. Мови логічного програмування.              | 6  | 2  | 2  | 2  |   | Звіт по лаб. роботі/2 |
| Тема 2. Вступ до мови Prolog. Синтаксичні особливості та вбудовані предикати.        | 6  | 2  | 2  | 2  |   | Звіт по лаб. роботі/2 |
| Тема 3. Факти та правила. Списки.  | 6  | 2  | 2  | 2  |   | Звіт по лаб. роботі/2 |
| Тема 4. Рекурсія. Рекурсивне представлення даних.                                    | 10 | 2  | 2  | 4  | 2 | Звіт по лаб. роботі/4 |
| Тема 5. Методи організації повторень мовою Prolog.                                   | 6  | 2  | 2  | 2  |   | Звіт по лаб. роботі/4 |
| Разом за модулем 2   | 34 | 10 | 10 | 12 | 2 | <b>14</b>             |
| <b>Види підсумкових робіт</b>  |    |    |    |    |   | <b>Бал</b>            |
| Тестування   |    |    |    |    |   | 30                    |
| Модульна контрольна робота 1   |    |    |    |    |   | 10                    |
| Модульна контрольна робота 2   |    |    |    |    |   | 10                    |

|                              |     |    |    |    |   |     |
|------------------------------|-----|----|----|----|---|-----|
| Модульна контрольна робота 3 |     |    |    |    |   | 10  |
| <b>Всього годин/Балів</b>    | 128 | 34 | 34 | 54 | 6 | 100 |

\*Форма контролю: ДС – дискусія, ДБ – дебати, Т – тести, ІР – індивідуальна робота здобувача освіти, РМГ – робота в малих групах, ТЗ – творче завдання, ПР – практична робота, РП – робота над проектом, РГ – рольова гра.

### Самостійна робота

**Самостійна робота здобувача є основним засобом засвоєння навчального матеріалу в час, вільний від обов'язкових занять, без участі викладача.**

#### Семестр 1

|  |        |
|--|--------|
| Опрацювання лекційного матеріалу.  | 5 год  |
| Перевірка здійснюється під час лабораторних занять та оцінюється при виставленні оцінки за змістовий модуль. |        |
| Підготовка до лабораторних занять, виконання домашніх завдань.   | 6 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.  | 21 год |
| Перевірка здійснюється під час модульних контрольних заходів і оцінюється відповідною кількістю балів.       |        |

| № з/п | Тема   | Кількість годин |
|-------|--|-----------------|
| 1     | Типи даних у C++ для представлення часу та дати.                     | 4               |
| 2     | Макроси у мові C++.  | 4               |
| 3     | Організація форматowanego виводу. Бібліотека <stdio>.                | 2               |
| 4     | Порозрядні оператори (I, АБО, НЕ).                                   | 4               |
| 5     | Специфікатори типів даних.   | 2               |
| 6     | Механізм використання покажчиків на структури і оператора "стрілка". | 4               |
|       | <b>Всього</b>  | 21              |

#### 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.   | 20 год |
| Перевірка здійснюється під час лабораторних занять.  |        |
| Систематизація вивченого матеріалу перед іспитом.  | 8 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     | Середовище StudioCode та робота з ним.   | 4               |
| 4     | Середовище CodeCombat та робота з ним.   | 4               |
| 5     | Середовищ Blockly та робота з ним.       | 4               |
| 6     | Середовищ Tynker та робота з ним.        | 4               |
|       | <b>Всього</b>                            | <b>24</b>       |

### Семестр 3

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 практичних занять.

Систематизація вивченого матеріалу перед контрольними заходами, тестуванням.

10 год

Перевірка здійснюється під час контрольних заходів, тестування.

Вивчення тем, що не розглядаються в курсі лекцій.

14 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     | Дата та час у Python. Форматування дат (Date Formatting).     | 2               |
| 3     | Вбудовані засоби для пошук та сортування даних в мові Python. | 4               |
| 4     | Абстрактні класи в мові Python.                               | 4               |
|       | <b>Всього</b>   | <b>14</b>       |

### Семестр 4

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.

18 год

Перевірка здійснюється під час лабораторних занять.

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

8 год

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

18 год

| № 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     | Робота з динамічною базою даних.                               | 4               |
| 5     | Мова програмування DataLog. Особливості та сфери застосування. | 2               |
|       | <b>Всього</b>  | <b>18</b>       |

## 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++.

## 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. Поняття інкапсуляції. Поняття про захист внутрішніх даних об'єкту у мові C++.
6. Конструктори, деструктори, їх роль та призначення.
7. Наслідування. Механізм наслідування членів-даних та методів у мові C++.
8. Доступ до членів класу та наслідування у мові C++.
9. Відношення клас-підклас та його зв'язок з відношенням абстрактне- конкретне. Сумісність типів знизу вгору.
10. Особливості виклику конструктора базового класу з конструктору надкласу. Порядок виклику конструкторів та деструкторів для об'єктів похідних класів.
11. Ієрархія класів. Множинне наслідування та його проблеми.
12. Поліморфізм та віртуальні функції. Поняття оголошеного та фактичного типу
13. Механізм виклику віртуальної функції.
14. Дружні функції, окремі класи та класи в цілому, обхід механізмів захисту членів класу.
15. Переваги та недоліки використання механізму дружності в програмах з об'єктно-орієнтованою композицією.
16. Перевантаження операторів функціями та методами.

Приклад завдання для іспиту:

1. 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 заліку

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 в Python: списки, кортежі, словники, рядкові величини, множини.
6. Функції, лямбда функції та рекурсія.
7. Робота з файлами в Python.
8. Модулі в Python.
9. Особливості оголошення класів у мові Python.
10. Застосування абстрактних класів у мові Python.
11. Поняття інкапсуляції. Поняття про захист внутрішніх даних об'єкту у мові Python.
12. Конструктори, деструктори, їх роль та призначення.
13. Наслідування. Механізм наслідування членів-даних та методів у мові Python.
14. Доступ до членів класу та наслідування у мові Python.
15. Особливості виклику конструктора базового класу з конструктору надкласу. Порядок виклику конструкторів та деструкторів для об'єктів похідних класів.
16. Ієрархія класів. Множинне наслідування та розв'язання diamond problem у мові Python.
17. Поліморфізм та віртуальні функції. Поняття оголошеного та фактичного типу у мові Python.
18. Механізм виклику віртуальної функції у мові Python.
19. Дружні функції, окремі класи та класи в цілому, обхід механізмів захисту членів класу у мові Python.

20. Переваги та недоліки використання механізму дружності в програмах з об'єктно-орієнтованою композицією у мові Python.
21. Перевантаження операторів функціями та методами у мові Python.
22. Віртуальні методи у мові Python.
23. Шаблони функцій та класів у мові Python.
24. Організація захищеного доступу у мові Python.
25. Декоратори у мові Python.

## Семестр 4

### Питання для іспиту

1. Парадигма функціонального програмування. Мова Scala.
2. Мова Scala. Типи даних (елементарні та складені).
3. Мова Scala. Функції. Типи функцій, каррінг.
4. Визначення функцій рівняннями. Зіставлення зі зразком. Приклади.
5. Визначення функцій рівняннями. Рекурсивні функції. Приклади.
6. Мова Scala. Інфіксні оператори та бінарні функції. Операторна форма бінарних функцій.
7. Особливості мови Haskell.
8. Парадигма логічного програмування. Мова Prolog.
9. Мова Prolog. Операційна семантика. Приклади.
10. Вступ до мови Prolog. Синтаксичні особливості та вбудовані предикати.
11. Мова Prolog. Факти та правила.
12. Мова Prolog. Рекурсія. Рекурсивне представлення даних.
13. Методи організації повторень мовою Prolog.
14. Метод відсікання та метод повернення в логічному програмуванні.
15. Мова Prolog. Списковий тип. Приклади.
16. Приклади використання мови Prolog (задача про 8 ферзів, родинні зв'язки).
17. Що таке JVM? Як вона використовується в програмах Java?
18. Основні типи даних в Java.
19. Які модифікатори доступу існують в Java і як вони використовуються?
20. Що таке поліморфізм і як його можна реалізувати в Java?
21. Які принципи ООП підтримуються в Java? Наведіть приклади.
22. Як використовувати наслідування і композицію в Java?

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).

## Політика курсу

### Політика щодо оцінювання

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 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 – 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.

Повторне складання екзамену допускається не більше як два рази: один раз – викладачеві, другий – комісії, яку створює декан факультету. Відповідно до п. 3.3 «Положення про визнання результатів навчання, отриманих у формальній, неформальній та/або інформальній освіті у Волинському національному університеті імені Лесі Українки» визнання результатів навчання у першому семестрі вивчення ОК «Програмування» не проводиться.

**Поєднання навчання та досліджень.** Здобувачі вищої освіти мають можливість додатково отримати бали за виконання індивідуальних завдань дослідницького характеру, зокрема, написання та опублікування наукових тез та статей з тематики дисципліни. За рішенням кафедри здобувачам, які брали участь у роботі конференцій, підготовці наукових публікацій, конкурсах студентських наукових робіт за тематикою ОК й досягли значних результатів, може бути присуджено додаткові (бонусні) бали, які зараховуються як результати поточного контролю. Систему бонусних балів погоджує науково-методична комісія факультету інформаційних технологій і математики. При цьому загальна кількість балів, що вноситься до відомості за поточну роботу, у випадку ОК, де передбачено екзамен, не може перевищувати 40 б.

### **Критерії оцінювання навчальних досягнень під час аудиторних занять**

#### **Високий рівень (100-76% максимальної оцінки)**

- Код працює на всіх тестових випадках.
- Відповідає вимогам завдання та заданим критеріям.
- Висока якість коду, ясна структура та ефективні алгоритми.
- Дотримання кращих практик програмування.
- Здобувач робить самостійні висновки про отримані результати.
- Здобувач для виконання завдання самостійно опрацьовує додаткові джерела, аналізує та інтерпретує матеріал.

- Здобувач відповідає на додаткові питання, які вимагають опрацювання додаткової літератури, коментує програмний код.

#### **Середній рівень (75-51% максимальної оцінки)**

- Код працює на більшості тестових випадків.
- Загальне виконання завдання, але можливі деякі невірності чи недоліки.
- Прийнятна якість коду, але існують можливості для вдосконалення.
- Здобувач використовує репродуктивний рівень засвоєння матеріалу, використовує додаткові джерела, які демонструють основні прийоми розв'язування задач або побудови алгоритмів.
- Здобувач відповідає на питання, які вимагають опрацювання конспекту лекцій, коментує програмний код.
- Здобувач після зауваження самостійно виправляє помилки у алгоритмі розв'язування задачі або у програмному коді.

#### **Низький рівень (50-26% максимальної оцінки)**

- Код майже не відповідає вимогам завдання.
- Значні помилки, відсутність деяких функцій.
- Погана структура коду та відсутність коментарів.
- Здобувач відповідає на питання, які вимагають опрацювання конспекту лекцій, усно коментує програмний код.
- Здобувач після зауваження самостійно виправляє помилки у алгоритмі розв'язування задачі або у програмному коді.

#### **Незадовільний рівень (25-0% максимальної оцінки)**

- Код та алгоритм майже або повністю не працює.
- Суттєві помилки у алгоритмі розв'язування задачі або у програмному коді.
- Відсутність зусиль у виконанні завдання.
- Здобувач лише після детальних інструкцій виправляє алгоритм розв'язування задачі або його реалізацію мовою програмування.
- Здобувач відповідає на питання, які вимагають опрацювання конспекту лекцій, не коментує програмний код.

### **Шкала оцінювання знань здобувачів освіти з формою контролю – екзамен**

| 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                    |

### **Вирішення конфліктних ситуацій**

Будь-яка конфліктна ситуація, яка виникає в учасників освітнього процесу вирішується згідно з Положенням про порядок і процедури вирішення конфліктних ситуацій у ВНУ імені Лесі Українки

### **Політика викладача щодо здобувача освіти**

Усі учасники освітнього процесу повинні дотримуватись вимог чинного законодавства України, Статуту і Правил внутрішнього розпорядку ВНУ імені Лесі Українки, загально-

прийнятих 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 та списування. Очікується, що всі здобувачі освіти відвідають усі лекції і практичні заняття освітнього компонента. Кожен здобувач повинен бути учасником дистанційних курсів “Алгоритмізація та програмування”, розміщених на платформі дистанційного навчання Moodle:

[https://moodle-cs.vnu.edu.ua/course/view.php?id=87,](https://moodle-cs.vnu.edu.ua/course/view.php?id=87)

[https://moodle-cs.vnu.edu.ua/course/view.php?id=86,](https://moodle-cs.vnu.edu.ua/course/view.php?id=86)

[https://moodle-cs.vnu.edu.ua/course/view.php?id=145,](https://moodle-cs.vnu.edu.ua/course/view.php?id=145)

[https://moodle-cs.vnu.edu.ua/course/view.php?id=139\).](https://moodle-cs.vnu.edu.ua/course/view.php?id=139)

### **Політика щодо академічної доброчесності**

Політика, стандарти та процедури дотримання академічної доброчесності у Волинському національному університеті імені Лесі Українки знайшли своє відображення в «Кодексі академічної доброчесності Волинського національного університету імені Лесі Українки». Вимоги до академічної доброчесності визначаються «Положенням про систему запобігання та виявлення академічного плагіату у науково-дослідній діяльності здобувачів вищої освіти і науково-педагогічних працівників Волинського національного університету імені Лесі Українки».

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 правдивої 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.

### **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, виконують всі завдання для аудиторних занять, всі домашні завдання. Прозвітуватися про виконання завдань можна під час консультацій, одночасно при цьому з'ясувати незрозумілі моменти, поставити запитання викладачу. Перекладання модульних контрольних робіт заборонено.

### **Рекомендовані інформаційні джерела**

#### **Семестр 1**

1. Гришанович Т. О. Алгоритмізація та програмування 1 [Електронний ресурс] : електронний курс навчальної дисципліни, затверджений НМР ВНУ імені Лесі Українки, протокол № 2 від 19.10.2022. ВНУ ім. Лесі Українки, 2022. URL: <https://moodle-cs.vnu.edu.ua/course/view.php?id=87>
2. Гришанович Т. О. Алгоритмізація та програмування 2 [Електронний ресурс] :

електронний курс навчальної дисципліни, затверджений НМР ВНУ імені Лесі Українки, протокол № 2 від 19.10.2022. ВНУ ім. Лесі Українки, 2022. URL: <https://moodle-cs.vnu.edu.ua/course/view.php?id=86>

3. Ефективне використання новітніх методів програмування графіки на С++ в навчальних цілях. / Л.Я. Глинчук, Т.О. Гришанович О.І. Кузьмич, Н.В. Багнюк. *COMPUTER-INTEGRATED TECHNOLOGIES: EDUCATION, SCIENCE, PRODUCTION*. 2020. № 40. С. 104–109. DOI: <https://doi.org/10.36910/6775-2524-0560-2020-40-16>
4. Розробка інтерактивної карти з використанням технології SPA. / А. О. Трофімук, Л. В. Булатецька, Ю. С. Павленко, Т. О. Гришанович. *COMPUTER-INTEGRATED TECHNOLOGIES: EDUCATION, SCIENCE, PRODUCTION*. 2021. № 42. С. 104–109. DOI: <https://doi.org/10.36910/6775-2524-0560-2021-42-15>.

## 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 данні (1 файл: 998 КБ). Луцьк : ВНУ імені Лесі Українки, 2022. 120 с. URI: <https://evnuir.vnu.edu.ua/handle/123456789/20320>

## Семестр 3

1. Костюченко А. О. Основи програмування мовою Python навчальний посібник. Чернігів : ФОП Баликіна С.М, 2020. 176 с.
2. Татарчук Д. Д., Діденко Ю. В. Об'єктно-орієнтоване програмування мовою Python конспект лекцій. Київ : КПІ ім. Ігоря Сікорського, 199. 128 с.
3. Юрченко І. В., Сікора В. С. Програмування мовою Python: навчальний посібник. Чернівці : Чернів. нац. ун-т., 2022. 104 с.
4. Фрімен Е. Head First. Патерни проєктування. Київ : Фабула, 2020. 672 с.

## Семестр 4

1. Гришанович Т. О. Функціональне програмування [Електронний ресурс] : електронний курс навчальної дисципліни. ВНУ ім. Лесі Українки, 2021. URL: <http://cs.vnu.edu.ua/moodle/course/view.php?id=102>
2. Заяць В. М. Логічне і функціональне програмування. Системний підхід : підручник / В. М. Заяць, М. М. Заяць. Рівне : НУВГП, 2018. 422 с
3. 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.
4. Документація Scala. URL: <https://docs.scala-lang.org/uk/>