Чернівецький національний університет імені Юрія Федьковича
Факультет математики та інформатики
Кафедра прикладної математики та інформаційних технологій

“ЗАТВЕРДЖУЮ”
Декан факультету
математики та інформатики

_____ Ольга МАРТИНЮК

“ 28 ” серпня 2026 року

РОБОЧА ПРОГРАМА
навчальної дисципліни

Програмування
(назва навчальної дисципліни)
Обов’язкова
(вказати: обов’язкова / вибіркова)
Освітньо-професійна програма Технології програмування та комп’ютерне моделювання
(назва програми)
Спеціальність F1 Прикладна математика
(вказати: код, назва)
Галузь знань F Інформаційні технології

Рівень вищої освіти перший (бакалаврський)
(вказати: перший бакалаврський/другий магістерський)
Факультет математики та інформатики
(назва факультету/інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання українська
(вказати: на якій мові читається дисципліна)

Чернівці – 2026

Робоча програма навчальної дисципліни "*Програмування"* **складена відповідно до** освітньо-професійної програми "Технології програмування та комп'ютерне моделювання" та навчального плану підготовки здобувачів вищої освіти за спеціальністю F1 Прикладна математика (галузь знань F Інформаційні технології). Дисципліна реалізує другу частину обов'язкового освітнього компонента ОК6 "Програмування" (загальний обсяг компонента - 13,0 кредитів ЄКТС) в обсязі 6,5 кредитів ECTS (195 годин).

Розробник:

Скутар Ігор Дмитрович, кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук

Викладач, що забезпечує читання даної навчальної дисципліни:

Скутар Ігор Дмитрович, кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук

Погоджено з гарантом ОП _____ **Василь МАЦЕНКО**
(підпис)

Затверджено на засіданні кафедри прикладної математики та інформаційних технологій

Протокол № 2 від « 28 » серпня 2026 року

Завідувач кафедри _____ **Ярослав БІГУН**
(підпис)

Схвалено методичною радою факультету математики та інформатики

Протокол № 1 від « 28 » серпня 2026 року

Голова методичної ради _____ **Тоня ФРАТАВЧАН**
(підпис)

Мета навчальної дисципліни:

- формування теоретичної бази знань студентів щодо побудови алгоритмів та їх реалізації мовою програмування;
- розвиток уміння розв'язувати алгоритмічні задачі, користуючись відомими теоретичними положеннями, математичним апаратом, літературою та комп'ютерною технікою;
- доведення вивчення курсу до творчого рівня;
- розвиток логічного, аналітичного мислення та основних видів розумової діяльності: уміння використовувати індукцію, дедукцію, аналіз, синтез, робити висновки, узагальнення;
- бачення студентами можливостей використання набутих знань у їх майбутній професії;
- інтеграція курсу з іншими дисциплінами, що викладаються в навчальному закладі.

Пререквізити: базові знання з математики, інформатики.

Результати навчання:

У результаті вивчення дисципліни студенти повинні ***знати*** основні поняття та твердження з програмного матеріалу даного курсу; ***вміти*** використовувати вивчений матеріал при розв'язуванні конкретних задач, застосовувати теоретичні знання на практиці.

Знання, які студент отримає в результаті вивчення даної дисципліни, відіграватимуть важливу роль у процесі його професійного формування та зростання, а також вони є необхідною основою для вивчення деяких інших фахових дисциплін.

Під час вивчення дисципліни, відповідно до освітньо-професійної програми, формуються наступні

загальні компетентності:

ЗК01 - здатність учитися і оволодівати сучасними знаннями;
ЗК02 - здатність застосовувати знання у практичних ситуаціях;
ЗК03 - здатність генерувати нові ідеї (креативність);
ЗК04 - здатність бути критичним і самокритичним;
ЗК06 - здатність до абстрактного мислення, аналізу та синтезу;
ЗК07 - здатність до пошуку, оброблення та аналізу інформації з різних джерел;
ЗК08 - знання та розуміння предметної області та розуміння професійної діяльності;
ЗК10 - навички у використанні інформаційних і комунікаційних технологій;
ЗК12 - визначеність і наполегливість щодо поставлених завдань і взятих обов'язків;
ЗК13 - навички міжособистісної взаємодії;

фахові компетентності:

ФК02 - здатність виконувати завдання, сформульовані у математичній формі;
ФК04 - здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію;

ФК06 - здатність розв’язувати професійні задачі за допомогою комп’ютерної техніки, комп’ютерних мереж та Інтернету, в середовищі сучасних операційних систем, з використанням стандартних офісних додатків;

ФК07 - здатність експлуатувати та обслуговувати програмне забезпечення автоматизованих та інформаційних систем різного призначення;

ФК08 - здатність використовувати сучасні технології програмування та тестування програмного забезпечення;

ФК10 - здатність створення документів встановленої звітності, використання нормативно-правових документів;

ФК17 - здатність до використання новітніх інформаційно-комунікаційних технологій;

ФК19 - здатність застосовувати сучасні інформаційні технології для комп’ютерної реалізації математичних моделей і алгоритми комп’ютерної графіки для розробки графічного програмного забезпечення;

та отримуються наступні **програмні результати навчання**:

ПРН11 - вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів;

ПРН14 - виявляти здатність до самонавчання та продовження професійного розвитку;

ПРН15 - уміти організувати власну діяльність та одержувати результат у рамках обмеженого часу;

ПРН18 - ефективно спілкуватися з питань інформації, ідей, проблем та рішень зі спеціалістами та суспільством загалом.

Опис навчальної дисципліни

Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні	
Денна	**1**	**1**	**6.5**	**195**	**30**	**30**	**0**	**30**	**105**	**0**	**залік**

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	
Змістовий модуль 1.												
Тема 1. Основні поняття алгоритмізації та програмування.	9	2	0	0		7						
Тема 2. Мова програмування Python.	9	2	0	0		7						
Тема 3. Логічні оператори і вирази.	17	2	4	4		7						
Тема 4. Циклічні конструкції.	17	2	4	4		7						
Тема 5. Одновимірні масиви. Найпростіші алгоритми роботи з одновимірними масивами.	15	2	2	4		7						
Тема 6. Упорядкування масивів.	17	2	4	4		7						
Тема 7. Двовимірні масиви.	13	2	2	2		7						

Тема 8. Базові алгоритми для обробки елементів двовимірних масивів.	13	2	2	2		7						
Тема 9. Словники.	13	2	2	2		7						
Разом за змістовим модулем 1	**123**	**18**	**20**	**22**		**63**						
Змістовий модуль 2.												
Тема 1. Рядки і символьні масиви. Стандартні функції для роботи з рядками.	13	2	2	2		7						
Тема 2. Найпростіші алгоритми роботи із символьними та рядковими величинами.	15	2	4	2		7						
Тема 3. Функції користувача.	15	2	4	2		7						
Тема 4. Модулі.	11	2	0	2		7						
Тема 5. Робота із файлами.	9	2	0	0		7						
Тема 6. Математичні пакети.	9	2	0	0		7						
Разом за змістовим модулем 2	**72**	**12**	**10**	**8**		**42**						
Усього годин за семестр	**195**	**30**	**30**	**30**		**105**						

Тематика лекційних занять з переліком питань

№ з/п	Назва теми	Кількість годин
1	Основні поняття алгоритмізації та програмування. • Основні поняття алгоритмізації та програмування. • Способи представлення алгоритмів. Мови програмування. • Базові алгоритмічні структури. Типи алгоритмів. Властивості алгоритмів. • Етапи розв'язування задач на комп'ютері.	2
2	Мова програмування Python. • Історія Python, переваги та інтегроване середовище IDLE. • Змінні, оператор присвоювання. • Числа та математичні оператори. • Введення даних та вивід результатів. • Модуль math. • Приклади програм лінійних алгоритмів.	2
3	Логічні оператори і вирази. • Логічні вирази. • Операції порівняння. • Логічні операції.	2

	• Оператор розгалуження if ... else. • Приклади програм розгалужених алгоритмів.	
4	Циклічні конструкції. • Оператор циклу while • Оператор циклу for • Перехід на наступну ітерацію циклу. Оператор continue • Переривання циклу. Оператор break • Приклади програм циклічних алгоритмів.	2
5	Одновимірні масиви. Найпростіші алгоритми роботи з одновимірними масивами. • Одновимірні масиви. • Списки та кортежі. • Найпростіші алгоритми роботи з одновимірними масивами.	2
6	Упорядкування масивів. • Масиви у Python. Модуль array. • Основні поняття алгоритмів сортування. • Сортування за допомогою вибору (by selection). • Сортування за допомогою обмінів (by exchange). • Сортування за допомогою включення (by insertion).	2
7	Двовимірні масиви. • Двовимірні списки. • Створення вкладених списків.	2
8	Базові алгоритми для обробки елементів двовимірних масивів. • Введення двовимірного масиву. • Обробка і виведення вкладених списків. • Пошук мінімального елемента. • Квадратні матриці.	2
9	Словники. • Асоціативні масиви. • Створення словників. • Доступ до елемента словника. • Методи словників.	2
10	Рядки і символьні масиви. Стандартні функції для роботи з рядками. • Рядки і символьні масиви. • Зміна регістру символів у рядках. • Конкатенація. • Операції над рядками. • Операція форматування. • Функції для роботи з символами	2
11	Найпростіші алгоритми роботи із символьними та рядковими величинами. • Методи для роботи з рядками. • Пошук та заміна у рядку.	2
12	Функції користувача. • Створення функцій. • Повернення із функції. • Параметри.	2

	• Області видимості. • Функції-генератори. • Анонімні функції. • Рекурсивні функції.	
13	Модулі. • Що таке модуль? • Простір імен і області видимості. • Імпорт модулів. • Стандартна бібліотека Python.	2
14	Робота із файлами. • Відкриття файла. • Зчитування даних з текстових файлів. • Запис у текстовий файл. • Інструкція with.	2
15	Математичні пакети. • Поняття про пакети. • Математичні пакети для Python.	2
	Разом	30

Тематика семінарських занять

Семінарські заняття не передбачені навчальним планом.

Тематика практичних занять

№ з/п	Назва теми	Кількість годин
1	Логічні оператори і вирази.	4
2	Циклічні конструкції.	4
3	Одновимірні масиви. Найпростіші алгоритми роботи з одновимірними масивами. Упорядкування масивів.	6
4	Двовимірні масиви. Алгоритми для обробки елементів двовимірних масивів.	4
5	Словники	2
6	Символьні рядки	6
7	Створення і використання функцій	4
	Разом	30

Тематика лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Лінійне програмування та умовний оператор	4
2	Циклічні програми	4
3	Алгоритми обробки масивів. Одновимірні масиви	4
4	Алгоритми обробки масивів. Багатовимірні масиви	4

5	Упорядкування і пошук даних	4
6	Словники	2
7	Символьні рядки	4
8	Створення і використання функцій	4
	Разом	30

Самостійна робота студентів

№ з/п	Назва теми	Кількість годин
1	Основні поняття алгоритмізації та програмування.	7
2	Мова програмування Python.	7
3	Логічні оператори і вирази.	7
4	Циклічні конструкції.	7
5	Одновимірні масиви. Найпростіші алгоритми роботи з одновимірними масивами.	7
6	Упорядкування масивів.	7
7	Двовимірні масиви.	7
8	Базові алгоритми для обробки елементів двовимірних масивів.	7
9	Словники.	7
10	Рядки і символьні масиви. Стандартні функції для роботи з рядками.	7
11	Найпростіші алгоритми роботи із символьними та рядковими величинами.	7
12	Функції користувача.	7
13	Модулі.	7
14	Робота із файлами.	7
15	Математичні пакети.	7
	Разом	105

Індивідуальні науково-дослідні завдання (ІНДЗ)

Навчальним планом та робочою програмою з навчальної дисципліни «Програмування» виконання ІНДЗ не передбачено.

Методи навчання

Під час викладання навчальної дисципліни «Програмування» застосовується комплексний підхід до вибору методів та освітніх технологій, спрямований на формування як ґрунтовного теоретичного фундаменту з алгоритмізації, так і

стійких практичних навичок розробки, відлагодження та тестування програмного забезпечення мовою програмування Python.

Освітній процес поєднує такі класичні та сучасні методи й освітні технології:

- **Пояснювально-ілюстративний метод:** викладач пояснює теоретичний матеріал, супроводжуючи пояснення прикладами коду та наочними ілюстраціями. Це допомагає студентам краще зрозуміти абстрактні концепції.
- **Репродуктивний метод:** студенти повторюють типові задачі та алгоритми, які розглядаються на заняттях. Це дозволяє їм закріпити базові навички програмування.
- **Проблемний метод:** викладач пропонує задачі, які потребують творчого підходу та аналізу. Студенти вчаться шукати оптимальні рішення, експериментуючи з різними підходами.
- **Метод проєктів:** студенти працюють над створенням власного проєкту на Python, що може включати розробку веб-додатку, автоматизацію процесів або аналіз даних.

Система контролю та оцінювання, критерії оцінювання поточного та підсумкового контролю

Формами поточного контролю є усна (захист лабораторних робіт) чи письмова (тестування, практичні завдання) відповіді студента тощо.

Формами підсумкового контролю є залік.

Поточний контроль*:** контрольне тестування після кожного модуля 1, практична контрольна робота, вибіркове опитування на лекційних заняттях. ***Підсумковий контроль – залік.

Критерії оцінювання результатів навчання з навчальної дисципліни: підсумковий модуль-контроль відбувається у вигляді тестування. Тест (30 балів) складається з 30-х теоретичних та практичних питань по 1 балу за кожну правильну відповідь. У сумі з модульними контролями (70 балів) це загалом становитиме 100 балів.

Підсумкова оцінка: Підсумкова оцінка виставляється за загальною сумою набраних студентом балів згідно із наступною таблицею:

<table>
<tr><td colspan="2">Поточне тестування та самостійна робота</td><td>Підсумковий тест (залік)</td><td>Сума</td></tr>
<tr><td>Змістовий модуль 1</td><td>Змістовий модуль 2</td><td rowspan="2">30</td><td rowspan="2">100</td></tr>
<tr><td>35</td><td>35</td></tr>
</table>

Бали у змістових модулях включають:

1. Лабораторні роботи – 40 балів
2. Модуль-контроль (тести) – 10 балів
3. Контрольна робота (практична) – 10 балів
4. Практичні заняття – 10 балів

Нормативно-процесуальні умови оцінювання, допуску та апеляції

Оцінювання знань здобувачів вищої освіти здійснюється відповідно до «Положення про організацію освітнього процесу в Чернівецькому національному університеті імені Юрія Федьковича» та «Положення про контроль і систему оцінювання результатів навчання здобувачів вищої освіти у Чернівецькому національному університеті імені Юрія Федьковича».

Умови допуску до підсумкового семестрового контролю (заліку):

- Обов'язкове виконання, подання та успішний захист усіх 8 лабораторних робіт, передбачених робочою програмою;
- Проходження поточного комп'ютерного тестування;

Політика ліквідації заборгованостей та апеляції:

- Здобувачі освіти, які не були допущені до заліку або отримали під час семестрового контролю незадовільну оцінку FX (35–49 балів), мають право на ліквідацію академічної заборгованості за затвердженим графіком у визначені терміни до початку наступного навчального семестру.
- Перескладання заліку дозволяється не більше двох разів: один раз — викладачу, вдруге – комісії, створеній деканом факультету. Перескладання семестрового контролю з метою підвищення позитивної оцінки не допускається.
- Здобувачі, які отримали оцінку F (1-34 бали), зобов'язані пройти повторний курс вивчення дисципліни.
- У разі незгоди з отриманою на семестровому заліку оцінкою здобувач вищої освіти має право подати письмову апеляційну заяву на ім'я декана факультету (або ректора Університету) не пізніше 12:00 наступного робочого дня після оголошення результату відповідно до Положення про апеляцію на результати підсумкового семестрового контролю ЧНУ.

Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	**Оцінка за шкалою ЄКТС**	
	Оцінка (бали)	**Пояснення за розширеною шкалою**
Відмінно	A (90-100)	відмінно
Добре	B (80-89)	дуже добре
	C (70-79)	добре
Задовільно	D (60-69)	задовільно
	E (50-59)	достатньо
Незадовільно	FX (35-49)	(незадовільно) з можливістю повторного складання
	F (1-34)	(незадовільно) з обов’язковим самостійним опрацюванням освітнього компоненту до перескладання

Зарахування результатів неформальної освіти

Порядок та умови визнання й зарахування результатів навчання, здобутих шляхом неформальної та/або інформальної освіти, регламентуються чинним у Чернівецькому національному університеті імені Юрія Федьковича «Порядком визнання у Чернівецькому національному університеті імені Юрія Федьковича результатів навчання, здобутих шляхом неформальної та/або інформальної освіти».

Здобувачі вищої освіти мають право на визнання результатів навчання, отриманих під час проходження онлайн-курсів (на платформах Coursera, edX, Prometheus, Udemy, Stepik, Pluralsight тощо), участі у воркшопах, хакатонах, літніх/зимових школах з програмування за наявності підтверджуючих документів (сертифікатів, свідоцтв).

Механізм та умови визнання результатів:

1. **Перезарахування окремих тем / лабораторних робіт / поточного контролю:**
   - У разі часткової відповідності змісту онлайн-курсу або сертифікованої програми тематиці окремих змістових модулів дисципліни «Програмування», здобувач звертається в усній формі до викладача, подаючи підтверджуючі документи.
   - За умови підтвердження відповідності результатів навчання здобувачу зараховується максимальний бал з відповідної теми (лабораторної роботи, тесту) та звільняється від виконання відповідних завдань. Загальний обсяг балів, який може бути зарахований здобувачу за результатами неформальної/інформальної освіти з окремих тем, не може перевищувати 25% від загальної кількості балів з дисципліни (до 25 балів зі 100).
2. **Повне зарахування освітнього компонента (підсумкового контролю):**
   - У разі необхідності повного зарахування навчальної дисципліни здобувач подає письмову заяву на ім’я декана факультету, до якої додає освітню декларацію про попереднє навчання та сертифікати.
   - Розгляд документів, співставлення результатів навчання із програмою дисципліни та оцінювання здійснює Предметна комісія, призначена розпорядженням декана.
3. **Процедура підтвердження та оцінювання:**
   - У разі прийняття Предметною комісією рішення про проведення контрольного заходу (співбесіди, контрольного завдання), здобувачу надається до 10 робочих днів для підготовки за питаннями і завданнями, що виносяться на контроль.
   - Загальний обсяг визнаних результатів неформальної освіти за весь строк навчання за ОПП «Технології програмування та комп’ютерне моделювання» (галузь знань F «Інформаційні технології») не може перевищувати 35% від загального обсягу освітньої програми.

Рекомендована література

основна:

1. Караванова Т. П. Інформатика. Основи алгоритмізації та програмування (процедурне програмування). Базовий курс : навчальний посібник. Доповнене та виправлене видання. – Шепетівка : Аспект, 2005. – 250 с.
2. Караванова Т. П. Основи алгоритмізації та програмування : 777 задач з рекомендаціями та прикладами : навчальний посібник. Доповнене та виправлене видання. – Київ : Генеза, 2006. – 288 с.
3. Бріггс Дж. Р. Python для дітей. Веселий вступ до програмування. – Львів : Видавництво Старого Лева, 2017. – 400 с.
4. Маттес Е. Пришвидшений курс Python. – Львів : Видавництво Старого Лева, 2021. – 600 с.
5. Васильєв О. Програмування мовою Python. – Тернопіль : Навчальна книга – Богдан, 2019. – 504 с.
6. Руденко В. Д., Жугастров О. П. Інформатика. Основи алгоритмізації і програмування мовою Python. – Харків : Ранок, 2024. – 192 с.

допоміжна:

1. Lutz M. Learning Python. 6th ed. – Sebastopol : O'Reilly Media, 2025. – 1500 p.
2. Dawson M. Python Programming for the Absolute Beginner. 3rd ed. – Boston : Course Technology, 2013. – 576 p.
3. Bhargava A. Grokking Algorithms. 2nd ed. – Shelter Island : Manning Publications, 2024. – 368 p.
4. Downey A. B. Think Python: How to Think Like a Computer Scientist. 3rd ed. – Sebastopol : O'Reilly Media, 2024. – 328 p.
5. Ceder N. The Quick Python Book. 4th ed. – Shelter Island : Manning Publications, 2025. – 580 p.

Інформаційні ресурси

1. Python Software Foundation. Python Official Website [Електронний ресурс]. – Режим доступу: https://www.python.org/.
2. Python Software Foundation. Python 3 Documentation [Електронний ресурс]. – Режим доступу: https://docs.python.org/3/.
3. Python Software Foundation. Python Tutorial [Електронний ресурс]. – Режим доступу: https://docs.python.org/3/tutorial/.
4. Python Tutor. Visualize Python, Java, C, C++, and JavaScript Code Execution [Електронний ресурс]. – Режим доступу: https://pythontutor.com/.
5. Подоба В. Python 3: Курс молодого бійця [Електронний ресурс]. – Режим доступу: http://www.vitaliypodoba.com/tutorials/python3-beginners-course/.

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- «Кодекс академічної доброчесності та етики Чернівецького національного університету імені Юрія Федьковича»

  https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/kodeks-akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yuriia-fedkovycha/

- «Положення про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича»

  https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyiavlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/

- «Політика використання штучного інтелекту в Чернівецькому національному університету імені Юрія Федьковича»

  https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/

Використання здобувачами освіти інструментів штучного інтелекту (ШІ) під час виконання навчальних завдань:

- *допускається за умови* дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов'язаний зазначити факт і спосіб його застосування);
- *не допускається* використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- *додаткові обмеження чи вимоги* щодо використання ШІ можуть визначатися викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.