

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



“ЗАТВЕРДЖУЮ”

Декан факультету
математики та інформатики

Ольга МАРТИНЮК

“25” червня 2025 року

РОБОЧА ПРОГРАМА
навчальної дисципліни

Програмування (2 семестр)

(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 2025 рік

Робоча програма навчальної дисципліни «Програмування (2 семестр)» складена відповідно до освітньо-професійної програми “Технології програмування та комп’ютерне моделювання”.

Розробник: Данилюк Іван Михайлович, канд. фіз.-мат. наук, доцент кафедри прикладної математики та інформаційних технологій
(П.І.Б. авторів, посада, науковий ступінь, вчене звання)

Викладачі, що забезпечують читання даної навчальної дисципліни:
Данилюк Іван Михайлович, канд. фіз.-мат. наук, доцент кафедри прикладної математики та інформаційних технологій
(П.І.Б. авторів, посада, науковий ступінь, вчене звання)
Унгурян Галина Михайлівна, канд. фіз.-мат. наук, асистент кафедри прикладної математики та інформаційних технологій
(П.І.Б. авторів, посада, науковий ступінь, вчене звання)

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

Затверджено на засіданні кафедри ПМІТ
Протокол № 13 від « 24 » червня 2025 року

Завідувач кафедри  **Ярослав БІГУН**
(підпис)

Схвалено
методичною радою факультету математики та інформатики
Протокол № 12 від « 25 » червня 2025 року

Голова методичної ради  **Віра СІКОРА**
(підпис)

Мета навчальної дисципліни полягає у розвитку логічного та аналітичного мислення студентів, уміння застосовувати індукцію, дедукцію, аналіз і синтез під час розв'язування завдань. Вона спрямована на формування ґрунтовної теоретичної бази знань щодо побудови алгоритмів та їх реалізації мовою програмування. Курс передбачає розвиток практичних навичок розв'язування алгоритмічних задач із використанням математичного апарату, спеціалізованої літератури та сучасної комп'ютерної техніки. Вивчення дисципліни має довести студентів до творчого рівня оволодіння матеріалом і сформуванню бачення шляхів застосування здобутих знань у майбутній професійній діяльності. Окрім цього, курс інтегрується з іншими навчальними дисциплінами, забезпечуючи цілісність підготовки фахівців. Таким чином, мета дисципліни полягає у гармонійному поєднанні теоретичних знань і практичних умінь, необхідних для формування компетентного спеціаліста.

Пререквізити: базові знання з математики, інформатики.

Результати навчання: у результаті вивчення навчальної дисципліни студент буде

знати:

- правила запису розробленого алгоритму мовою програмування С;
- можливості використання середовища компіляції розробленої С-програми;
- загальні принципи розв'язування алгоритмічних задач; постановка задачі, побудова алгоритму, реалізація алгоритму мовою програмування, тестування реалізованого алгоритму;
- методи розв'язання алгоритмічних задач.

вміти:

- застосовувати теоретичні знання для розв'язування практичних завдань;
- використовувати навички роботи з інтегрованим середовищем програмування;
- аналізувати відомі методи побудови алгоритмів та визначати найоптимальніші з них для розв'язування конкретної задачі;
- реалізовувати побудовані алгоритми мовою програмування С;
- розробляти власні тести для перевірки коректності розроблених алгоритмів;
- тестувати розроблені алгоритми;
- використовувати навички техніки програмування.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів компетентностей, передбачених відповідним стандартом вищої освіти України (за наявності) або ОП, програмні результати навчання якої відповідають вимогам Національної рамки кваліфікацій (за відсутності стандарту):

- загальних:

- 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;

- фахових:

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.

Результати вивчення дисципліни деталізують такі програмні результати навчання ОП:

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	2	7	210	30	30		30	120		Іспит
Заочна	-	-	-	-	-	-	-	-	-	-	-

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин												
	денна форма						Заочна форма						
	усього го	у тому числі					усього го	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	13	
Змістовий модуль 1.													
Тема 1. Мова програмування С. Основи.	10	2	2	2		4							
Тема 2. Логічні оператори і вирази.	12	2	2	2		6							
Тема 3. Циклічні конструкції.	14	2	2	2		8							
Тема 4. Алгоритми поєднання розгалуження та повторення. Вкладені цикли. Покрокове введення та виведення даних. Рекурентні послідовності.	14	2	2	2		8							
Разом за змістовим модулем 1	50	8	8	8		26							
Змістовий модуль 2.													
Тема 5. Одновимірні масиви. Найпростіші алгоритми роботи з одновимірними масивами. Пошук заданого елемента, пошук мінімального/максимального елемента.	14	2	2	2		8							
Тема 6. Додаткові способи введення даних. Основи роботи з файлами і потоками. Генератор випадкових чисел. Константні вхідні дані, ініціалізація масивів.	14	2	2	2		8							
Тема 7. Двовимірні масиви. Базові алгоритми для обробки елементів двовимірних масивів.	14	2	2	2		8							
Тема 8. Рядки і символічні масиви. Стандартні функції для	16	2	2	2		10							

роботи з рядками.													
Тема 9. Найпростіші алгоритми роботи із символічними та рядковими величинами.	14	2	2	2		8							
Разом за змістовим модулем 2	72	10	10	10		42							
Змістовий модуль 3.													
Тема 10. Вказівники. Операції з вказівниками. Практичне застосування вказівників.	20	2	4	4		10							
Тема 11. Використання вказівників для роботи з масивами. Динамічні масиви.	18	2	2	2		12							
Тема 12. Функції користувача. Масиви і функції. Передача масиву у функцію.	14	2	2	2		8							
Тема 13. Рекурсивні алгоритми.	12	2	2	2		6							
Тема 14. Структури. Масиви структур. Структури в структурах.	12	2	1	1		8							
Тема 15. Використання структур для роботи з функціями. Використання вказівників для роботи зі структурами.	12	2	1	1		8							
Разом за змістовим модулем 3	88	12	12	12		52							
Усього годин	210	30	30	30		120							

Тематика лекційних занять з переліком питань

№	Назва теми з основними питаннями
1.	<p>Мова програмування С. Основи</p> <ol style="list-style-type: none"> 1. Поняття алгоритму та програми 2. Різниця між void main() і int main() 3. Структура програми на С 4. Змінні та типи даних 5. Базові операції над змінними 6. Інкрементні та декрементні операції 7. Коментарі в кодї 8. Форматований ввід/вивід 9. Особливості написання коду на С 10. Практичні приклади
2.	<p>Логічні оператори і вирази. Розгалуження</p> <ol style="list-style-type: none"> 1. Логічні оператори та булева логіка в С 2. Операції відношення 3. Логічні операції 4. Пріоритет операцій 5. Особливості булевих виразів у С 6. Умовний вираз (?:) 7. Умовний оператор (if, else) 8. Вкладені умови 9. Оператор switch 10. Практичні приклади 11. Порівняння з іншими мовами

3.	Циклічні конструкції <ol style="list-style-type: none"> 1. Поняття циклу 2. Цикл із лічильником (for) 3. Цикли за умовою 4. Вкладені цикли 5. Практичні приклади 6. Порівняння з іншими мовами
4.	Алгоритми поєднання розгалуження та повторення. Вкладені цикли. Покрокове введення та виведення даних. Рекурентні послідовності <ol style="list-style-type: none"> 1. Поєднання розгалужень із циклами 2. Вкладені цикли 3. Поєднання з розгалуженнями 4. Покрокове введення та виведення даних 5. Використання while для невизначеної кількості даних 6. Рекурентні послідовності 7. Практичні приклади 8. Порівняння з іншими мовами
5.	Одновимірні масиви. Найпростіші алгоритми роботи з одновимірними масивами. <ol style="list-style-type: none"> 1. Пошук заданого елемента, пошук мінімального/максимального елемента 2. Поняття масиву 3. Оголошення та ініціалізація масивів 4. Базові операції з масивами 5. Найпростіші алгоритми роботи з одновимірними масивами 6. Розширені приклади роботи з масивами 7. Порівняння з іншими мовами 8. Рекомендації щодо використання
6.	Додаткові способи введення даних <ol style="list-style-type: none"> 1. Робота з файлами та потоками 2. Генерація випадкових чисел 3. Ініціалізація масивів константами 4. Практичні приклади 5. Порівняння з іншими мовами
7.	Двовимірні масиви. Базові алгоритми для обробки елементів двовимірних масивів <ol style="list-style-type: none"> 1. Поняття двовимірного масиву 2. Оголошення та ініціалізація двовимірних масивів 3. Введення та виведення двовимірних масивів 4. Базові алгоритми обробки двовимірних масивів 5. Порівняння з іншими мовами
8.	Рядки і символні масиви. Стандартні функції для роботи з рядками <ol style="list-style-type: none"> 1. Поняття рядка в мові C 2. Основні операції з рядками 3. Стандартні функції бібліотеки <string.h> 4. Практичні приклади 5. Порівняння з іншими мовами
9.	Найпростіші алгоритми роботи з рядками. Вступ до алгоритмів роботи з рядками <ol style="list-style-type: none"> 1. Основи алгоритмів із рядками 2. Практичні приклади
10.	Вказівники. Операції з вказівниками. Практичне застосування вказівників <ol style="list-style-type: none"> 1. Оператор адреси (&)

	<ul style="list-style-type: none"> 2. Оператор розіменування (*) 3. Вказівники: оголошення та основи 4. Присвоювання значень вказівнику 5. Розіменування вказівників 6. Розіменування некоректних вказівників 7. Розмір вказівників. Залежність від архітектури 8. Практичне застосування вказівників 9. Виділення та звільнення пам'яті 10. Додаткові приклади 11. Порівняння з іншими мовами
11.	Використання вказівників для роботи з масивами. Динамічні масиви <ul style="list-style-type: none"> 1. Вказівники і масиви: схожість і відмінності 2. Передача масивів у функції 3. Адресна арифметика і індексація масивів 4. Динамічні масиви 5. Практичні приклади
12.	Функції користувача <ul style="list-style-type: none"> 1. Основи функцій: параметри та аргументи 2. Способи передачі аргументів 3. Повернення значень із функцій 4. Практичні приклади
13.	Рекурсивні алгоритми <ul style="list-style-type: none"> 1. Стек і купа: організація пам'яті в С 2. Рекурсія: основи та принципи 3. Рекурсивні алгоритми: приклади 4. Рекурсія vs Ітерація 5. Додаткові приклади
14.	Структури <ul style="list-style-type: none"> 1. Навіщо потрібні структури? 2. Оголошення та визначення структур 3. Доступ до членів структур 4. Ініціалізація структур 5. Вкладені структури 6. Розмір структур і вирівнювання
15.	Використання структур для роботи з функціями. Використання вказівників для роботи зі структурами <ul style="list-style-type: none"> 1. Передача структур у функції 2. Повернення структур із функцій 3. Використання вказівників зі структурами 4. Вкладені структури та вказівники 5. Практичні приклади

Тематика практичних занять з переліком питань

№	Назва теми (питання/завдання)
1, 2	Лінійне програмування та умовний оператор
3, 4	Циклічні програми
5, 6	Алгоритми обробки масивів
7, 8	Символьні рядки

9	Організація роботи з файлами
10 - 12	Вказівники. Динамічний розподіл пам'яті
13	Створення і використання функцій
14	Рекурсивні підпрограми
15	Структуровані типи даних

Тематика лабораторних занять з переліком питань

№	Назва теми (питання/завдання)
1	<p>Лінійне програмування та умовний оператор</p> <p>Лінійне програмування та умовний оператор</p> <ol style="list-style-type: none"> 1. Засвоєння структури програми та візуального проектування програм. 2. Засвоєння простих типів даних, операцій та виразів. 3. Засвоєння основних операторів мови програмування (присвоєння, складеного, умовної і безумовної передачі керування, вибору). 4. Отримання навичок в організації введення та виведення даних. 5. Отримання практичних навичок роботи в інтерактивному режимі.
2	<p>Циклічні програми</p> <ol style="list-style-type: none"> 1. Засвоєння операторів циклу. 2. Отримання практичних навичок розробки ітераційних алгоритмів і програм для наближеного обчислення нескінченних сум та пошуку членів у нескінченних послідовностях, що задовольняють певні умови.
3	<p>Алгоритми обробки масивів</p> <ol style="list-style-type: none"> 1. Засвоєння структурованих даних типу масив. 2. Отримання практичних навичок розробки традиційних обчислювальних алгоритмів обробки масивів даних. 3. Засвоєння операторів циклу та побудови програм з використанням вкладених циклів. 4. Отримання практичних навичок розробки традиційних обчислювальних алгоритмів обробки багатовимірних масивів.
4	<p>Упорядкування і пошук даних</p> <ol style="list-style-type: none"> 1. Засвоєння методів упорядкування і пошуку даних. 2. Отримання практичних навичок розробки алгоритмів і програм упорядкування і пошуку даних.
5	<p>Організація роботи з файлами</p> <p>Символьні рядки</p> <ol style="list-style-type: none"> 1. Засвоєння структурованих даних типу символьний рядок і операцій над цими даними. 2. Отримання практичних навичок обробки і редагування текстів.
6	<p>Вказівники. Динамічний розподіл пам'яті</p> <ol style="list-style-type: none"> 1. Ознайомитися з концепцією вказівників та їх використанням у мові програмування С для роботи з пам'яттю. 2. Освоїти методи динамічного розподілу пам'яті за допомогою функцій <code>calloc</code>, <code>malloc</code> та звільнення пам'яті за допомогою <code>free</code>. 3. Навчитися застосовувати два способи доступу до елементів масиву при динамічному виділенні пам'яті: через індексацію <code>[]</code> та через оператор розмінування <code>*</code>. 4. Поглибити розуміння роботи з одновимірними масивами та стрічками при використанні вказівників. 5. Розвинути навички оптимізації програмного коду шляхом ефективного управління пам'яттю.
7	<p>Створення і використання функцій</p> <ol style="list-style-type: none"> 1. Ознайомлення з поняттям функції у програмуванні.

	<p>2. Дослідження особливостей локальних і глобальних змінних та механізм повернення значень.</p> <p>3. Закріплення навичок структурованого програмування та оптимізації коду за допомогою функцій.</p>
8	<p>Рекурсивні підпрограми</p> <p>1. Ознайомлення з основними принципами рекурсії.</p> <p>2. Застосування рекурсивних алгоритмів, дослідження переваг та недоліків рекурсивного підходу в порівнянні з ітеративним методом.</p> <p>3. Закріплення навичок програмування рекурсивних функцій та аналізу їх складності.</p>
9	<p>Структуровані типи даних</p> <p>1. Ознайомлення з поняттями структури у програмуванні.</p> <p>2. Вивчення принципів їх використання для організації та зберігання даних.</p> <p>3. Створення, ініціалізування та робота із структурами.</p> <p>4. Дослідження переваг у порівнянні з іншими типами даних.</p> <p>5. Закріплення навичок роботи зі структурами через практичне виконання завдань.</p>

Завдання для самостійної роботи студентів

№	Назва теми	Кількість годин
1	Мова програмування C. Основи.	4
2	Логічні оператори і вирази.	6
3	Циклічні конструкції.	8
4	Алгоритми поєднання розгалуження та повторення. Вкладені цикли. Покрокове введення та виведення даних. Рекурентні послідовності.	8
5	Одновимірні масиви. Найпростіші алгоритми роботи з одновимірними масивами. Пошук заданого елемента, пошук мінімального/максимального елемента.	8
6	Додаткові способи введення даних. Основи роботи з файлами і потоками. Генератор випадкових чисел. Константні вхідні дані, ініціалізація масивів.	8
7	Двовимірні масиви. Базові алгоритми для обробки елементів двовимірних масивів.	8
8	Рядки і символьні масиви. Стандартні функції для роботи з рядками.	10
9	Найпростіші алгоритми роботи із символьними та рядковими величинами.	8
10	Вказівники. Операції з вказівниками. Практичне застосування вказівників.	10
11	Використання вказівників для роботи з масивами. Динамічні масиви.	12
12	Функції користувача. Масиви і функції. Передача масиву у функцію.	8
13	Рекурсивні алгоритми.	6
14	Структури. Масиви структур. Структури в структурах.	8
15	Використання структур для роботи з функціями. Використання вказівників для роботи зі структурами.	8
	Разом	120

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
80-89	B	добре	
70-79	C		
60-69	D	задовільно	
50-59	E		
35-49	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Перелік питань для самоконтролю та підсумкового контролю

1. Поняття алгоритму та програми.
2. Різниця між void main() і int main().
3. Структура програми мовою C.
4. Змінні та типи даних у C.
5. Базові операції над змінними.
6. Інкрементні та декрементні операції.
7. Використання коментарів у програмному кодї.
8. Форматований ввід та вивід у C.
9. Особливості написання коду мовою C.
10. Логічні оператори та булева логіка у C.
11. Операції відношення та їх використання.
12. Пріоритет логічних і арифметичних операцій.
13. Умовний вираз ?:.
14. Умовні оператори if, else, вкладені умови.
15. Оператор switch та його застосування.
16. Поняття циклу та його різновиди.
17. Цикл із лічильником (for).
18. Цикли за умовою (while, do...while).
19. Вкладені цикли та їх застосування.
20. Поєднання розгалужень і циклів.
21. Покрокове введення та виведення даних.
22. Використання циклів для обробки невизначеної кількості даних.
23. Рекурентні послідовності та способи їх реалізації.
24. Масиви: поняття, оголошення та ініціалізація.
25. Базові операції з одновимірними масивами.
26. Пошук заданого елемента в масиві.

27. Пошук мінімального та максимального елементів.
28. Алгоритми обробки масивів (сортування, пошук).
29. Додаткові способи введення даних у C.
30. Основи роботи з файлами та потоками.
31. Генерація випадкових чисел у програмах.
32. Ініціалізація масивів константами.
33. Поняття двовимірного масиву.
34. Оголошення та ініціалізація двовимірних масивів.
35. Введення та виведення елементів двовимірного масиву.
36. Базові алгоритми обробки двовимірних масивів.
37. Поняття рядка в мові C.
38. Символьні масиви та їх використання.
39. Основні функції бібліотеки <string.h>.
40. Алгоритми роботи з рядками (пошук, конкатенація, порівняння).
41. Практичні приклади роботи з текстами.
42. Поняття вказівника та його призначення.
43. Оператор адреси & та оператор розіменування *.
44. Оголошення вказівників та операції з ними.
45. Розіменування та некоректні вказівники.
46. Розмір вказівників залежно від архітектури.
47. Динамічний розподіл пам'яті (malloc, calloc, free).
48. Використання вказівників для роботи з масивами.
49. Адресна арифметика та індексація.
50. Передача масивів у функції.
51. Динамічні масиви: створення та використання.
52. Поняття функції у програмуванні.
53. Параметри та аргументи функцій.
54. Способи передачі аргументів у функцію.
55. Повернення значень із функцій.
56. Масиви та функції: передача масиву у функцію.
57. Використання рекурсії у програмуванні.
58. Рекурсія та ітерація: порівняння.
59. Приклади рекурсивних алгоритмів.
60. Стек і купа: організація пам'яті.
61. Структури: поняття, оголошення та ініціалізація.
62. Доступ до елементів структур.
63. Масиви структур та вкладені структури.
64. Використання структур у функціях.
65. Використання вказівників для роботи зі структурами.
66. Алгоритми сортування (методи впорядкування даних).
67. Алгоритми пошуку у структурованих даних.
68. Обробка текстів і символьних рядків.
69. Організація роботи з файлами: читання та запис.
70. Використання динамічного розподілу пам'яті для оптимізації коду.
71. Програмування та відлагодження ітераційних алгоритмів.

72. Програмування рекурсивних підпрограм.
73. Організація даних у структурованому вигляді.
74. Порівняння основних можливостей мови С з іншими мовами програмування.
75. Тестування та відлагодження програм у середовищі розробки.

Рекомендована література

1. Електронний курс
<https://moodle.chnu.edu.ua/course/view.php?id=3450>
2. Алгоритми і структура даних: Навчальний посібник / В.М.Ткачук. - ІваноФранківськ : Видавництво Прикарпатського національного університету імені Василя Стефаника, 2016. 286 с.
3. Алгоритми та структури даних. Навчальний посібник / Т. О. Коротєєва. Львів : Видавництво Львівської політехніки, 2014. - 280 с.
4. Ковалюк Т.В. Основи програмування. / Ковалюк Т.В. Київ: ВНУ Київ, 2005. 400 с.
5. Цвіркун Л.І. Розробка програмного забезпечення комп'ютерних систем. Програмування: навч. посіб. / Л.І. Цвіркун, А.А. Євстігнєєва, Я.В. Панферова; під заг. ред. Л.І. Цвіркуна; М-во освіти і науки України, Нац. гірн. ун-т. – 3-тє вид., випр. – Дніпропетровськ: НГУ, 2016. – 223 с
6. Шпак З.Я. Програмування мовою С / З.Я. Шпак. – Львів: Оріяна-Нова, 2006. – 432 с.
7. Белов Ю. А., Карнаух Т. О., Коваль Ю. В., Ставровський А. Б. Вступ до програмування мовою С++. Організація обчислень : навч. посіб./ Ю. А. Белов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. – К. : Видавничо-поліграфічний центр “Київський університет”, 2012. – 175 с.
8. Бичков О.С. Основи сучасного програмування [Текст] : підручник / О. С. Бичков ; Київ. нац. ун-т ім. Т. Шевченка. – К. : Київ. ун-т, 2008. – 272 с.
9. Ковалюк Т.В. Алгоритмізація та програмування: Підручник. – Львів: «Магнолія 2006», 2013. – 400 с.
10. Роберт Мартін. Чистий код – Фабула, 2019. – 416с.
11. Brian “Beej Jorgensen” Hall. Beej’s Guide to C Programming. – 2013
https://beej.us/guide/bgc/pdf/bgc_a4_c_1.pdf
12. Mike Vanahan, Declan Brady, Mark Doran. The C Book
http://publications.gbdirect.co.uk/c_book/
13. Steve Oualline. C Elements of Style
<http://www.oualline.com/books.free/style/index.html>
14. Jens Gustedt. Modern C
http://icube-icps.unistra.fr/img_auth.php/d/db/ModernC.pdf
15. Brian Gough. An Introduction to GCC
<http://www.network-theory.co.uk/docs/gccintro/>

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні навчальної дисципліни регламентовано такими документами:

- Етичний кодекс Чернівецького національного університету імені Юрія Федьковича
- Положення про виявлення та запобігання академічного плагіату у Чернівецькому національному університету імені Юрія Федьковича