

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ВСП «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ, ПРАВА ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ЗУНУ»**



ЗАТВЕРДЖУЮ

Директор ВСП «ФКЕПІТ ЗУНУ»

Василь МАРТИНЮК

«31» серпня 2023р.

**РОБОЧА ПРОГРАМА
З ДИСЦИПЛІНИ
«Програмування»**

Освітньо-професійний ступінь – **фаховий молодший бакалавр**

Галузь знань: **12 Інформаційні технології**

Спеціальність: **123 Комп'ютерна інженерія**

Освітньо-професійна програма: **«Комп'ютерна інженерія»**

Циклова комісія комп'ютерної інженерії

Форма навчання	Курс	Семестр	Лекції (год.)	Практичні (год.)	Самостійна робота	Загальний обсяг, годин	Екзамен/ залік
Денна	3	5,6	60	60	90	210	Залік/Екзамен

Тернопіль – 2023 р.

Робоча програма складена на основі освітньо-професійної програми підготовки фахового молодшого бакалавра галузі знань **12 Інформаційні технології**, спеціальності **123 Комп'ютерна інженерія**, затвердженої Педагогічною радою ВСП «ФКЕПІТ ЗУНУ» «30» серпня 2021р.

Робоча програма складена викладачем Мельник В.О.

Робоча програма затверджена на засіданні циклової комісії комп'ютерної інженерії, протокол № 1 від 28 серпня 2023р.

Голова циклової комісії



Сергій МАРКОПОЛЬСЬКИЙ

**СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«Програмування»**

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, освітньо-професійна програма, освітньо-професійний ступінь	Характеристика навчальної дисципліни
		<i>Денна форма навчання</i>
Змістових модулів – 6	Галузь знань: 12 Інформаційні технології	Нормативна
	Спеціальність: 123 Комп'ютерна інженерія	
Загальна кількість годин - 210	Освітньо-професійна програма «Комп'ютерна інженерія» Освітньо-професійний ступінь: фаховий молодший бакалавр	Лекції (60 год.)
		Лабораторні (60 год.)
		Самостійна робота (90 год.)
		Вид контролю: залік/екзамен

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мета: навчальної дисципліни «Програмування» є оволодіння основами сучасних мов та технологій програмування, набуття умінь та досвіду зі створення прикладних програм, програмних комплексів з врахуванням особливостей сучасних мов та технологій програмування при вирішенні задач у науковій, інженерно-технічній та економічній сферах діяльності та сприяння розвитку логічного та аналітичного мислення студентів.

Завдання: формування теоретичних знань та практичних навичок у майбутніх фахівців відповідно до поставленої мети; вивчення основних етапів процесу проектування програмного забезпечення і визначення принципів процедурного програмування щодо розробки програм мовою C++; вивчення типових підходів до розробки і аналізу найбільш розповсюджених алгоритмів рішення типових задач; здійснення аналізу можливостей сучасних інструментальних середовищ розробки програм; побудова базових алгоритмів пошуку та сортування, передавання та опрацювання різних типів даних; формування знань, вмінь і навичок ефективного використання засобів програмування у своїй майбутній професійній діяльності; формування у студентів основи інформаційної культури.

У результаті вивчення навчальної дисципліни «Програмування» студент повинен:

знати:

- основи мови програмування C++ та технологію створення програм;
- основні етапи процесу проектування програмного забезпечення;
- типові алгоритмічні конструкції;
- принципи процедурного і структурованого програмування;
- особливості застосування сучасних базових інструментальних програмних засобів, призначених для вирішення типових задач;
- базові типи даних;
- похідні типи даних: переліки, покажчики, посилання, масиви, структури, об'єднання;
- оператори управління програмою;
- правила роботи з функціями;
- систему ведення-виведення C++;
- основні принципи роботи з файлами.

вміти:

- виконувати формалізацію постановки задачі та її математично точний опис;
- складати програми мовою C++, забезпечуючи: рішення задач обробки даних; створення і обробку структур, масивів структур; найпростішу обробку файлів; використання функцій; використання сучасного інструментального програмного забезпечення;
- користуватися раніше складеними програмами і здійснювати супровід програм, вносити зміни в програму,
- виконувати відлагодження програм за допомогою вбудованих інструментальних засобів обґрунтовувати прийняті рішення.

Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни «Програмування»

- СК2. Здатність застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування апаратних, програмних та інструментальних засобів комп'ютерної інженерії.

- СК4. Здатність брати участь у розробці системного та прикладного програмного забезпечення засобів комп'ютерної інженерії з використанням ефективних алгоритмів, сучасних методів і мов програмування.

- СК10. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати прийняті рішення.

- СК12. Здатність створювати, впроваджувати, адмініструвати бази даних і знань з використанням сучасних методів, технологій та систем керування базами даних

Результати навчання:

- РН2. Знати і розуміти теоретичні положення, що лежать в основі функціонування апаратних та програмних засобів комп'ютерної інженерії.

- РН3. Знати сучасні методи та технології для розв'язання прикладних задач комп'ютерної інженерії.

- РН6. Тестувати, діагностувати та обслуговувати апаратні та програмні засоби комп'ютерної інженерії.

- РН7. Застосувати знання для формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

- РН8. Застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації апаратних та програмних засобів комп'ютерної інженерії для вирішення технічних задач у професійній діяльності.

- РН9. Розробляти, тестувати, впроваджувати, експлуатувати програмне забезпечення для вбудованих та розподілених систем.

- РН11. Ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів комп'ютерної інженерії.

- РН12. Поєднувати теорію і практику, знаходити та обґрунтовувати шляхи рішення типових задач у професійній діяльності з урахуванням виробничих інтересів.

- РН14. Використовувати сучасні інтегровані середовища, методи і технології розробки, впровадження, адміністрування комп'ютерних систем та мереж, без даних і знань.

- РН16. Спілкуватись усно та письмово з професійних питань українською та іноземними мовами.

- РН17. Вміти обґрунтовувати прийняті рішення, оцінювати, оформляти та представляти результати професійної діяльності згідно діючій нормативній документації.

- РН18. Вміти використовувати сучасні інтегровані середовища, методи і технології розробки, впровадження, адміністрування комп'ютерних систем та мереж, баз даних і знань.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «Програмування»

Змістовний модуль 1.

Тема 1. Апаратні та програмні засоби ЕОМ.

Ознайомлення з ЕОМ, виділення основних частин в програмні і апаратні.

Тема 2. Елементи мови С++.

Історія створення мови С++. Характеристика С-систем. Алфавіт і словник мови С++. Загальна структура програми на мові С++. Перша програма — “Hello world”. Поняття типу даних. Типи даних. Мінімальні відомості про базові типи. Директиви передпроцесора. Стандартні функції введення/виведення мови С++. Використання коментарів. Об'ява змінних. Загальні поняття щодо типів даних мови С. Числові типи та символічний тип. Загальні поняття щодо пам'яті комп'ютера. Руйнуюче та не руйнуюче зчитування. Арифметика в мові С++. Пріоритет виконання операцій. Алгоритми та програми. Мови програмування. Блок-схема як засіб графічного зображення алгоритмів. Лінійні алгоритми. Алгоритми з розгалуженням. Цикли та циклічні структури.

Тема 3. Файлова система. Системи числення.

Файлова система. Файл. Різновиди систем числення. Одинці інформації. Основні позиційні системи числення. Переведення чисел з однієї системи числення до іншої.

Змістовний модуль 2.

Тема 4. Елементи мови С++. Програмування базових алгоритмів.

Історія С++. Алфавіт мови С++. Типи даних. Константи. Правила записування арифметичних виразів. Операнди і вирази. Арифметичні операції. Оператори присвоювання. Зведення типів. Математичні функції. Види базових алгоритмів. Програмування лінійних алгоритмів. Програмування розгалужених алгоритмів. Програмування циклічних алгоритмів.

Тема 5. Масиви в С++.

Масиви як структурований тип даних. Визначення масивів. Синтаксис об'яви масивів. Поняття “елемент масиву” та “індекс масиву”. Синтаксис використання масивів. Зв'язок покажчиків з масивами. Вирази та арифметика з покажчиками. Динамічний розподіл пам'яті. Стандартні алгоритми роботи з масивами: визначення екстремальних значень, статистичних показників, упорядкування масивів, пошук визначеного елемента. Багатовимірні масиви. Приклади програм з багатовимірними масивами. Реалізація алгоритмів, пов'язаних з обробкою матриць.

Тема 6. Покажчики. Динамічна пам'ять.

Ідеологія використання покажчиків в програмі на мові С++. Об'ява та ініціалізація покажчиків. Приклади використання покажчиків. Операції з покажчиками. Виведення значень покажчиків як адрес оперативної пам'яті. Виведення значення, на яке посилається покажчик. Вирази та арифметика з

показчиками. Доступ до апаратних елементів комп'ютера за допомогою показників. Приклади програм з використанням показників.

Тема 7. Символи та рядки.

Огляд роботи зі символами, їхнє представлення та операції, які можна проводити з символами. Розуміння, що таке рядки та як вони використовуються для зберігання текстової інформації. Виконання операцій з рядками: Огляд основних операцій з рядками, таких як конкатенація, пошук, заміна символів та підрядків, також ознайомлення з внутрішніми функціями для роботи з рядками.

Тема 8. Двовимірні масиви в C++

Організація двовимірних масивів. Введення-виведення двовимірних масивів. Програмування базових алгоритмів опрацювання двовимірних масивів. Опрацювання двовимірних масивів у функціях.

Змістовний модуль 3.

Тема 9. Стандартні бібліотеки C++.

Потоковий ввід-вивід в програмах на C++. Ввід-вивід файлів. Стандартна бібліотека шаблонів (STL). Алгоритми, контейнери та ітератори.

Тема 10. Функції. Правила організації функцій.

Правила організації функцій. Способи передавання параметрів до функцій. Параметри зі значеннями за замовченням. Функції з різною кількістю параметрів.

Тема 11. Рекурсія. Перевантаження функцій.

Рекурсивні функції. Правила використання рекурсивних функцій. Перевантаження функцій.

Тема 12. Багатофайлові програми

Міжфайлова взаємодія. Заголовні файли. Бібліотеки функцій. Директиви препроцесора. Область дії та простір імен.

Змістовний модуль 4.

Тема 13. Типи користувача.

Перейменування типів. Структури. Об'єднання. Перерахування. Множини.

Тема 14. Об'єктно-орієнтований аналіз та проектування.

Поняття об'єктно-орієнтованого аналізу, проектування та програмування. Об'єктна модель предметного середовища, принципи її побудови. Основи об'єктно-орієнтованого проектування мовою UML. Канонічні діаграми UML. Основи об'єктно-орієнтованої мови програмування, доповнення C++ до C.

Тема 15. Технологія об'єктно-орієнтованого програмування мовою C++.

Класи і об'єкти в мові C++. Абстрагування та інкапсуляція. Перевантаження операцій та функцій. Перетворення типів, що визначаються класом. Наслідування та ієрархії класів. Просте та множинне наслідування. Реалізація поліморфізму: віртуальні функції та поліморфічні кластери. Области дії та простори імен.

Тема 16. Файли. Текстові файли.

Загальні відомості про файли. Текстові файли. Робота з текстовими файлами у стилі C. Робота з текстовими файловими потоками у стилі C++. Послідовне записування до файлу і зчитування з файлу. Довільне записування до файлу і зчитування з файлу. Опрацювання текстових файлів за допомогою дескрипторів.

Тема 17. Бінарні файли.

Робота з бінарними файлами у стилі C. Робота з бінарними файловими потоками у стилі C++. Опрацювання бінарних файлів за допомогою дескрипторів.

Змістовний модуль 5.

Тема 18. Модульне і об'єктно-орієнтоване програмування.

Підхід модульного програмування. Підхід об'єктно-орієнтованого програмування. Основні поняття ООП. Поняття поліморфізму. Поняття інкапсуляції. Поняття наслідування.

Тема 19. Класи

Оголошення класу. Визначення методів класу, тобто реалізації конкретних функцій-елементів даного класу. Специфікатори доступу.

Тема 20. Об'єкти. Конструктори.

Поняття об'єкту. Властивості конструкторів. Властивості конструкторів. Конструктор зі списком ініціалізації. Конструктор за замовчуванням. Конструктор копіювання.

Тема 21. Визначення методів класу поза класом. Деструктори.

Причини використання методів поза класом. Способи визначення методів поза класом. Деструктори. Причини використання деструкторів.

Тема 22. Перевантаження операцій

Правила перевантаження унарних операцій. Перевантаження унарних операцій. Перевантаження бінарних операцій. Перевантаження операції присвоювання. Перевантаження операції індексування.

Тема 23. Наслідування.

Переваги використання наслідування. Недоліки використання наслідування. Термінологія. Застосування. Типи наслідування.

Тема 24. Поліморфізм.

Типи поліморфізму. Приклади. Спеціалізований поліморфізм. Параметричний поліморфізм. Поліморфізм підтипів. Статичний та динамічний поліморфізм.

Змістовний модуль 6.

Тема 25. Налаштування програм.

Помилки компіляції. Попередження і підказки. Компонування. Помилки етапу виконання.

Тема 26. Сучасні технології програмування мовою С++.

Імперативне програмування. Функціональне програмування. Логічне програмування. Паралельне та конкурентне програмування. Декларативне програмування.

Тема 27. Візуальне середовище розробки програм мовою С++.

Технологія візуального програмування. Середовище Microsoft Visual С++ Express. Етапи розробки додатків

Тема 28. Інструменти візуальної розробки додатків.

Характеристика основних компонентів середовища. Компонент Button. Компонент Label. Компоненти CheckBox і RadioButton

Тема 29. Розробка графічного інтерфейсу користувача.

Компонент MenuStrip. Компонент ToolStrip. Компонент StatusStrip. Стандартні діалогові вікна

Тема 30. Графічні можливості мови С++.

Компонент PictureBox. Виведення графічних зображень. Олівці і пензлі. Графічні примітиви. Метод базової точки

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№	Назви змістових модулів і тем	КІЛЬКІСТЬ ГОДИН		
		Лекції	Практичні (лабораторні) заняття	Самостійна робота
1	2	3	4	5
Змістовний модуль 1				
1.	Тема 1. Апаратні та програмні засоби ЕОМ.	2	2	3
2.	Тема 2. Програмне забезпечення.	2	2	3
3.	Тема 3. Файлова система. Системи числення.	2	2	3
Змістовний модуль 2				
4.	Тема 4. Елементи мови С++. Програмування базових алгоритмів.	2	2	3
5.	Тема 5. Масиви в С++.	2	2	3

6.	Тема 6. Показчики. Динамічна пам'ять.	2	2	3
7.	Тема 7. Символи і рядки.	2	2	3
8.	Тема 8. Двовимірні масиви в C++	2	2	3
Змістовний модуль 3				
9	Тема 9. Стандартні бібліотеки C++.	2	2	3
10	Тема 10. Функції. Правила організації функцій.	2	2	3
11	Тема 11. Рекурсія. Перевантаження функцій.	2	2	3
12	Тема 12. Багатофайлові програми.	2	2	3
Змістовний модуль 4				
13	Тема 13. Типи користувача.	2	2	3
14	Тема 14. Об'єктно-орієнтований аналіз та проектування.	2	2	3
15	Тема 15. Технологія об'єктно-орієнтованого програмування мовою C++.	2	2	3
16	Тема 16. Файли. Текстові файли.	2	2	3
17	Тема 17. Бінарні файли.	2	2	3
Змістовний модуль 5				
18	Тема 18. Модульне і об'єктно-орієнтоване програмування.	2	2	3
19	Тема 19. Класи.	2	2	3
20	Тема 20. Об'єкти. Конструктори.	2	2	3
21	Тема 21. Визначення методів класу поза класом. Деструктори.	2	2	3
22	Тема 22. Перевантаження операцій.	2	2	3
23	Тема 23. Наслідування	2	2	3
24	Тема 24. Поліморфізм	2	2	3
Змістовний модуль 6.				
25	Тема 25. Налаштування програм.	2	2	3

26	Тема 26. Сучасні технології програмування мовою С++.	2	2	3
27	Тема 27. Візуальне середовище розробки програм мовою С++.	2	2	3
28	Тема 28. Інструменти візуальної розробки додатків.	2	2	3
29	Тема 29. Розробка графічного інтерфейсу користувача.	2	2	3
30	Тема 30. Графічні можливості мови С++.	2	2	3
	Разом	60	60	90

5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ з дисципліни «Програмування»

Практичне заняття 1

Тема 1. Алгоритми переводу чисел із однієї системи числення до іншої

План практичного заняття

1. Вступ до систем числення
2. Алгоритми переводу

Практичне заняття 2

Тема 2. Лінійні алгоритми та програми

План практичного заняття

1. Огляд лінійних алгоритмів
2. Практичні завдання

Практичне заняття 3

Тема 3. Програмування базових алгоритмів.

План практичного заняття

1. Огляд базових алгоритмів
2. Практичні завдання

Практичне заняття 4

Тема 4. Програмування розгалужених алгоритмів.

План практичного заняття

1. Огляд розгалужених алгоритмів
2. Практичні завдання

Практичне заняття 5.

Тема 5. Програмування циклічних алгоритмів.

План практичного заняття

1. Огляд циклічних алгоритмів

2. Практичні завдання

Практичне заняття 6

Тема 6. Одновимірні масиви

План практичного заняття

1. Огляд одновимірних масивів
2. Операції з масивами
3. Практичні завдання з масивами

Практичне заняття 7

Тема 7. Робота з стрічками.

План практичного заняття

1. Огляд стандартної бібліотеки для роботи з стрічками.
2. Практичні завдання з стрічками

Практичне заняття 8

Тема 8. Багатовимірні масиви

План практичного заняття

1. Багатовимірні масиви: огляд
2. Робота з багатовимірними масивами
3. Практичні завдання

Практичне заняття 9

Тема 6. Робота з динамічними масивами.

План практичного заняття

1. Робота з динамічним одновимірним масивом.
2. Робота з динамічним двовимірним масивом.

Практичне заняття 10

Тема 10. Програми з використанням підпрограм.

План практичного заняття

1. Підпрограми: визначення та призначення
2. Створення та використання підпрограм
3. Практичні завдання з підпрограмами

Практичне заняття 11

Тема 11. Рекурсія.

План практичного заняття

1. Поняття рекурсії.
2. Умови виходу з рекурсії.
3. Практичні завдання з використанням рекурсії.

Практичне заняття 12

Тема 12. Перезавантаження функцій.

План практичного заняття

1. Способи перезавантаження функцій.

2. Практичні завдання на перезавантаження функцій.

Практичне заняття 13

Тема 13. Створення програми з використанням декількох файлів.

План практичного заняття

1. Директиви підключення сторонніх файлів.
2. Директиви умовної компіляції.
3. Практичні завдання.

Практичне заняття 14

Тема 14. Робота з типами користувача користувача

1. Практичні завдання на перейменування типів.
2. Практичні завдання на роботу з структурами.

Практичне заняття 15

Тема 15. Робота з типами користувача користувача

1. Практичні завдання на роботу з об'єднаннями.
2. Практичні завдання на роботу з множинами.

Практичне заняття 16.

Тема 16. Робота з текстовими файлами.

1. Читання текстового файлу в стилі C.
2. Читання текстового файлу в стилі C++.
3. Запис у текстовий файл в стилі C.
4. Запис у текстовий файл в стилі C++.

Практичне заняття 17.

Тема 17. Робота з бінарними файлами.

1. Читання бінарного файлу в стилі C.
2. Читання бінарного файлу в стилі C++.
3. Запис у бінарний файл в стилі C.
4. Запис у бінарний файл в стилі C++.

Практичне заняття 18

Тема 18. Класи і об'єкти в мові C++.

План практичного заняття

1. Основи класів та об'єктів
2. Створення класів та об'єктів
3. Практичні завдання з використанням класів

Практичне заняття 19

Тема 19. Використання конструкторів.

План практичного заняття

4. Використання конструктора зі списком ініціалізації.
5. Використання конструктора за замовчуванням.
6. Використання конструктора копіювання.

Практичне заняття 20

Тема 20. Визначення методів класу поза класом

План практичного заняття

1. Способи визначення методів класу.
2. Поняття статичних методів.
3. Практичне використання.

Практичне заняття 21

Тема 10. Деструктори.

План практичного заняття

1. Поняття деструктора.
2. Перевантаження в практиці
3. Практичні завдання з перевантаженням

Практичне заняття 22

Тема 22. Перевантаження операцій.

План практичного заняття

1. Перевантаження унарних операцій.
2. Перевантаження бінарних операцій.
3. Перевантаження операції присвоювання.
4. Перевантаження операції індексування.

Практичне заняття 23

Тема 23. Наслідування та ієрархії класів.

План практичного заняття

1. Наслідування в ООП
2. Створення ієрархій класів
3. Практичні завдання з успадкуванням

Практична заняття 24

Тема 24. Реалізація поліморфізму: віртуальні функції.

План практичного заняття

1. Поліморфізм
2. Використання віртуальних функцій
3. Практичні завдання з поліморфізмом

Практичне заняття 25

Тема 25. Оброблення виняткових ситуацій. Множинне успадкування.

План практичного заняття

1. Виняткові ситуації: визначення та обробка
2. Розгляд виняткових ситуацій у програмуванні та їх обробка.

Практичне заняття 26

Тема 26. Параметричний поліморфізм. Шаблони функцій

План практичного заняття

1. Параметричний поліморфізм: використання шаблонів
2. Робота з шаблонами функцій
3. Практичні завдання з параметричним поліморфізмом

Практичне заняття 27

Тема 27. Параметричний поліморфізм. Шаблони функцій

План практичного заняття

1. Параметричний поліморфізм: використання шаблонів
2. Робота з шаблонами функцій

Практичне заняття 28

Тема 28. Побудова першого віконного застосунку

План практичного заняття

1. Вступ до віконних застосунків
2. Використання інтерфейсів
3. Практичне завдання: створення віконного застосунку

Практичне заняття 29

Тема 29. Побудова застосунку з використанням декількох вікон

План практичного заняття

1. Способи створення нових вікон.
2. Робота з контекстними вікнами.
3. Практичне завдання

Практичне заняття 30

Тема 30. Створення найпростішого графічного редактора.

План практичного заняття

1. Створення графічного вікна для редактора.
2. Робота з найпростішими примітивами.
3. Створення графічного редактора

6. САМОСТІЙНА РОБОТА

№ з/п	Назва теми
1	Історія розвитку мови програмування C++
2	Типізація даних в C++: вбудовані, похідні типи і класи.
3	Призначення і використання покажчиків в C++.
4	Масиви та їх реалізація в C++.
5	Операції, які використовуються в мові програмування C++. Пріоритети виконання операцій.
6	Конструкції мови програмування C++, що управляють.

7	Робота з рядками в C++.
8	Типові математичні функції мови C++.
9	Стандартні потоки вводу-виводу.
10	Файлові потоки вводу-виводу.
11	Динамічний розподіл і звільнення пам'яті.
12	Загальна характеристика динамічних структур даних.
13	Діаграми станів UML.
14	Діаграми діяльності UML.
15	Діаграми компонентів UML.
16	Діаграми розгортання UML.
17	Конфлікт імен при множинному успадкуванні.
18	Порядок виклику конструкторів при множинному успадкуванні.
19	Віртуальні базові класи.
20	Програмування мовою C++ з використанням шаблонів функцій.
21	Шаблони класів в програмах на C++
22	Ієрархії класів з використанням шаблонів.
23	Поліморфізм в програмах на C++.
24	Склад та можливості стандартних бібліотек мови C++
25	Контейнери STL: черга.
26	Контейнери STL: пріоритетна черга.
27	Контейнери STL: стек
28	Алгоритми стандартної бібліотеки шаблонів STL.

7. МЕТОДИ НАВЧАННЯ

Вивчення дисципліни передбачає: лекції, пояснення; демонстрація вимірювальної апаратури та вимірювальних трактів; ілюстрація схем вимірювальних трактів та спрямованих графічних схем; лабораторний метод, самостійне опрацювання програмного матеріалу, що не викладається на лекціях, практичні заняття, самостійну роботу студентів в тому числі з використанням сучасної комп'ютерної техніки; індивідуальні . У навчальному процесі застосовуються: лекції, в тому числі із використанням засобів комп'ютерної техніки; практичні заняття; індивідуальні заняття і виконання самостійної роботи,

виконання розрахунково-графічних завдань під керівництвом викладача і самостійно; робота в Інтернет, робота на платформах Zoom, Moodle.

Організація навчання здійснюється за кредитно-модульною системою з елементами тестування та рейтинговим оцінюванням знань студентів у відповідності з Концепцією впровадження в Україні Болонського процесу

8. ФОРМИ КОНТРОЛЮ

І семестр

Рубіжна атестація № 1	Директорська контрольна робота (№ 2)	Середній бал (№ 3)	Залік
25%	50%	25%	100%

II семестр

Рубіжна атестація (№ 1)	Директорська контрольна робота (№ 2)	Середній бал (№ 3)	Екзамен	Разом
20%	25%	15%	40%	100%

9. КРИТЕРІЇ ОЦІНЮВАННЯ

КРИТЕРІЇ ОЦІНЮВАННЯ НАВЧАЛЬНИХ ДОСЯГНЕНЬ СТУДЕНТІВ З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Оцінка за 100-бальною шкалою / ECTS	Оцінка за національною шкалою	Критерії оцінювання навчальних досягнень здобувачів освіти	Рівень компетентності
90-100 А	Відмінно	Оцінюється завдання, що містить відповіді, в яких навчальний матеріал відтворюється в повному обсязі. Здобувач освіти активно працює протягом усього курсу і показує при цьому високий рівень володіння теоретичними знаннями і практичними вміннями, відповідь його правильна, обґрунтована, повна, логічна, містить аналіз, систематизацію, узагальнення навчального матеріалу, здатний висловити власне ставлення до альтернативних міркувань з конкретної проблеми, проявляє вміння здійснювати зв'язок теоретичних занять з практичними і реалізувати міжпредметні зв'язки. Практичне завдання виконане правильно, як з використанням типового алгоритму, так і за самостійно розробленим алгоритмом.	Високий (творчий) рівень
85-89 В	Дуже добре	Оцінюється завдання, що містить відповіді в яких відтворюється значна частина навчального матеріалу. Здобувач освіти виявляє достатній рівень володіння теоретичним матеріалом і практичним вмінням з несуттєвими неточностями, певною мірою може аналізувати матеріал, порівнювати та робити висновки з окремих питань навчального матеріалу.	Достатній рівень

<p>75-84</p> <p>С</p>	<p>Добре</p>	<p>Оцінюється завдання, що містить відповіді, в яких відтворюється незначна частина навчального матеріалу. Здобувач освіти виявляє достатній рівень володіння теоретичним матеріалом і практичним вмінням з певними неточностями та недоліками, достатньо проявляє вміння самостійно та аргументовано викладати матеріал.</p>	
<p>65-74</p> <p>Д</p>	<p>Задовільно</p>	<p>Оцінюється завдання, що містить відповіді в яких виявляється середній рівень володіння теоретичними знаннями, розуміння навчального матеріалу, а також практичних вмінь. Здобувач освіти в цілому оволодів суттю питань з даної теми, намагається аналізувати факти та події, робити певні конкретні висновки, недостатньо проявляє вміння самостійно та аргументовано викладати матеріал, виявляє недостатній рівень поінформованості у практичному застосуванні.</p>	<p>Середній рівень</p>
<p>60-64</p> <p>Е</p>	<p>Достатньо</p>	<p>Оцінюється завдання, що містить відповіді, в яких відтворюються мінімальні знання і розуміння навчального матеріалу, проте недостатньо вони глибокі та осмислені. Здобувач освіти на заняттях поводить себе пасивно, відповідає лише за викликом викладача, дає неповні відповіді на запитання, частково аналізує навчальний матеріал, проте при цьому робить неконкретні та неточні висновки, виявляє низький рівень поінформованості у практичному застосуванні.</p>	
		<p>Оцінюється завдання, що не виконане або містить відповіді на</p>	

35-5 FX	Незадовільно	рівні елементарного відтворення окремих фактів, елементів, об'єктів, фрагментів навчального матеріалу. Здобувач освіти відтворює незначну частину навчального матеріалу, викладає його уривчастими реченнями, безсистемно, відсутні висновки, узагальнення. Під час відповіді здобувач освіти допускає суттєві помилки, які не здатен виправити після коригуючих запитань, рівень володіння теоретичними знаннями і практичним вмінням не задовольняє мінімальні критерії. Обов'язкове складання повторного екзамену, заліку.	Низький рівень
1-34 F	Незадовільно	Оцінюється завдання, що не виконане, у здобувача освіти відсутні елементарні як теоретичні знання, так і практичні вміння. Обов'язкове повторне вивчення навчальної дисципліни.	Незадовільний рівень

10. ШКАЛА ОЦІНЮВАННЯ

Оцінка за 100-бальною шкалою	Оцінка за національною шкалою		Оцінка за шкалою ECTS	Рівень компетентності
	Екзамен	Залік		
90-100	5 (відмінно)	Зараховано	A	Високий (творчий) рівень
85-89	4 (дуже добре)		B	Достатній рівень
75-84	4 (добре)		C	
65-74	3(задовільно)		D	Середній рівень
60-64	3 (достатньо)		E	
35-59 (незадовільний)	2 (незадовільно з можливістю повторного	Незараховано (з можливістю повторного	FX	Низький рівень

рівень)	складання екзамену)	складання заліку)		
1-34	2 (незадовільно з обов'язковим повторним вивченням дисципліни)	Незараховано (з обов'язковим повторним вивченням дисципліни)	X	Незадовільний рівень

11. МАТЕРІАЛЬНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

1. Тексти лекцій.
2. Завдання для лабораторних завдань.
3. Завдання для самостійної роботи.
4. Тести для перевірки знань студентів.
5. Засоби діагностики знань студентів.
6. Критерії оцінювання знань студентів.
7. Посібники.
8. Презентаційні матеріали.
9. Програмні продукти Dev-C++
10. Інтернет-джерела.

12. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Базова:

1. Arlow J. UML 2 & the Unified Process: Practical ObjectOriented Analysis and Design, 2nd Ed. / Jim Arlow, Ila Neustadt - Addison_Wesley, 2005.
2. Rumbaugh J. The Unified Modeling Language Reference Manual», 2nd Ed. / James Rumbaugh, Ivar Jacobson, Grady Booch - Addison_Wesley, 2005.
3. Jacobson I. Unified Software Development Process /Ivar Jacobson, Grady Booch, James Rumbaugh - Addison_Wesley, 1999.
4. Booch G. Object-Oriented Analysis and Design with Applications, 3 Ed. / Grady Booch - Addison-Wesley, 2007.
5. Booch G. Object Solutions / Grady Booch - Addison_Wesley, 1995.
6. Gamma E. Design Patterns / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides - Addison_Wesley, 1995.

Допоміжна:

1. Алексенко О.В. Технології програмування та створення програмних продуктів. Конспект лекцій. – Суми : Сумський державний університет, 2013. – 133 с.
2. Прийма С.М. Теорія алгоритмів: навчальний посібник. Мелітополь: ФОП Однорог Т. В., 2018. 116 с.
3. Шаховська Н. Б., Голощук Р. О. Алгоритми і структури даних. Навчальний посібник. Львів : Магнолія, 2018. 216 с.