

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІВАНО-ФРАНКІВСЬКИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
МЕНЕДЖМЕНТУ**

ЗАТВЕРДЖУЮ:
Директор ІФННІМ
Галина ЛЯХОВИЧ

«*Галина Ляхович*»

2024р.



ЗАТВЕРДЖУЮ:
Проректор з НПР
Віктор ОСТРОВЕРХОВ

«*Віктор Островерхов*» 2024 р.



РОБОЧА ПРОГРАМА

з дисципліни
«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

ступінь вищої освіти – бакалавр
галузь знань – 01 Освіта/Педагогіка
спеціальність – 015.39 Професійна освіта
освітньо-професійна програма – Професійна освіта
(Цифрові технології)

Кафедра управління та адміністрування

Форма навчання	Курс	Семестр	Лекції (год.)	Практ. (год.)	ІРС (год.)	Тренінг, (год.)	Самост. робота студ. (год.)	Разом (год.)	Іспит (сем.)
Денна	2	3	30	30	4	8	78	150	3

*30.08.2024р.
Віктор*

Івано-Франківськ, ІФННІМ – 2024

**СТРУКТУРА РОБОЧОЇ ПРОГРАМИ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»**

1. Опис дисципліни «Об'єктно-орієнтоване програмування»

Дисципліна – «Об'єктно-орієнтоване програмування»	Галузь знань, спеціальність, рівень вищої освіти, освітньо-професійна програма	Характеристика навчальної дисципліни
Кількість кредитів ECTS – 5	Галузь знань: 01 «Освіта/Педагогіка»	Статус дисципліни блок обов'язкових дисциплін, цикл професійної підготовки Мова навчання – українська
Кількість залікових модулів – 4	Спеціальність: 015.39 «Професійна освіта»	Рік підготовки: Денна – 2 Семестр: Денна – 3
Кількість змістових модулів – 2	Ступінь вищої освіти – бакалавр	Лекції: Денна – 30 год. Практичні заняття: Денна – 30 год.
Загальна кількість годин – 150		Самостійна робота: Денна – 78 год. Тренінг – 8 Індивідуальна робота Денна – 4 год.
Тижневих годин – 8 з них аудиторних – 2		Вид підсумкового контролю – екзамен

2. Мета й завдання вивчення дисципліни «ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

2.1. Мета вивчення дисципліни

Сформувати у студентів знання, вміння і навички, необхідні для усвідомлення і раціонального використання понять об'єктно-орієнтованого програмування. Метою викладання дисципліни є розкриття сучасних наукових концепцій, понять, методів та технологій об'єктно-орієнтованого аналізу, проектування та програмування на прикладі інформаційних управляючих систем конкретного призначення.

2.2. Завдання вивчення дисципліни

Основними завданнями вивчення дисципліни є:

- оволодіння концепціями, поняттями, методами та технологіями об'єктно-орієнтованого програмування;
- оволодіння методами та технологіями об'єктно-орієнтованого аналізу і проектування,
- оволодіння методами оброблення виключень та застосування бібліотек класів;
- оволодіння методами та засобами об'єктно-орієнтованого програмування windows-застосувань;

2.3. Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни

- здатність самостійно проектувати компоненти програмного забезпечення;
- здатність самостійно реалізувати прототипи архітектури програмного забезпечення;
- здатність самостійно установлювати, налаштовувати та обслуговувати системне, інструментальне та прикладне програмне забезпечення та інформаційні системи.
- здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.
- здатність обгрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення

2.4. Передумови вивчення дисципліни

Дисципліни, які повинні бути вивчені попередньо:

- «Основи інженерії програмного забезпечення»;
- «Основи програмування»;

2.5. Програмні результати навчання

В результаті вивчення дисципліни «Об'єктно-орієнтоване програмування» студент повинен знати:

- алгоритмічну мову C++ разом із засобами об'єктно-орієнтованого програмування в обсязі програми курсу;
- принципи проектування і програмування класів і застосування об'єктів;
- програмування структурних елементів класів;
- принципи успадкування класів і будову ієрархій класів;
- проектування шаблонів класів;
- структуру і застосування стандартної бібліотеки шаблонів STL.
- проектувати, кодувати, тестувати і налагоджувати програми C++ в обсязі всіх можливостей мови, будувати програмну взаємодію технології об'єктно-орієнтованого програмування з іншими методами і парадигмами;

- проектувати класи і використовувати технологію об'єктно-орієнтованого програмування в повному обсязі сучасних вимог;
- складати шаблони класів для різних типів даних відповідно до задач;
- застосовувати стандартні класи бібліотеки шаблонів STL.

3. Програма навчальної дисципліни «ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

Змістовий модуль 1. Теоретичні засади об'єктно-орієнтованого програмування

Тема 1. Вступ до об'єктно-орієнтованого програмування. Поняття класу та об'єкту.

Основні поняття об'єктно-орієнтованого програмування. Історія об'єктно-орієнтованого програмування. Необхідність використання об'єктно-орієнтованого підходу. Зв'язок між даними та методами їх обробки в об'єктно-орієнтованому програмуванні. Поняття: інкапсуляція, успадкування класів, поліморфізм. Відмінності між процедурним програмуванням та об'єктно-орієнтованим. Абстракція даних. Поняття об'єкта в ООП. Стан та поведінка об'єкта. Життєвий цикл об'єкта. Відношення між об'єктами. Поняття класу. Типи класів. Відношення між класами. Інстанціювання. призначення методу Get() та Set().

Тема 2. Конструктори та деструктори класу.

Класи, члени класу. Ініціалізація. Інтерфейс та реалізація. Конструктори і деструктори. Властивості деструктора Особливості конструктора. Кількість параметрів у конструкторі. Конструювання і знищення об'єктів. Конструктор копіювання і його властивості. Конструктор перетворення типів і його властивості. Вказівник this. Вказівники на члени класу. Статичні змінні класу. Статичні методи класу. Поняття дружності.

Тема 3. Робота із масивами об'єктів.

Масиви. Багатовимірні масиви. Ступінчасті (зубчасті) масиви масивів. Звернення до елемента масиву. Одновимірний масив об'єктів. Одновимірний динамічний масив об'єктів. Різниця між масивами об'єктів та масивами, що складаються із елементів простих типів. Ініціалізування масивів, які складається із об'єктів. Метод елемента масиву об'єктів. Оголошення двовимірного масиву об'єктів. Оголошення двовимірного динамічного масиву об'єктів.

Тема 4. Види класів. Вкладені класи.

Клас як основа об'єктно-орієнтованого програмування. Правила взаємовідношення між класами та їх розміщення у програмі. Глобальні класи, оголошення яких здійснюється поза функціями. Локальні класи, які розміщуються всередині функції; Дружні класи, яким надають права доступу до усіх частин класу. Ітератори, призначені для забезпечення доступу до елементів інших класів. Успадковані класи, коли похідний клас використовує оголошення даних та методів базового класу. Шаблонні класи, які є узагальненим визначенням множини класів. Абстрактні класи, що містять нереалізовані віртуальні методи. Інтерфейсні класи, призначені для однакового доступу до різних версій об'єктів (компонентів) відкомпільованого класу, незалежно від їх внутрішньої реалізації.

Тема 5. Перевантаження операторів.

Операції, що можна перевантажити. Синтаксис операторних функцій. Вимоги до операторних функцій. Обмеження, що накладаються на застосування перевантажених операторів. Форми перевантаження операторів інкремента і декремента. Особливості перевантаження операторів інкремента і декремента. Обмеження, що накладаються на функцію operator->(). Особливості бінарних операторних функцій-членів. Вимоги до перевантаженого оператору присвоювання. Вимоги до перевантаженого операторної функції operator,(). Перевантаження бінарного оператора доступу до члена масиву.

Тема 6. Успадкування. Одинарне успадкування.

Поняття повторного використання коду. Механізми, що дозволяють створювати ієрархії класів. Визначення базового класу. Особливості похідного класу. Поняття вивід класу. Види успадкування залежно від кількості базових класів. Механізми успадкування їх

значення. Види успадкування залежно від рівня доступу. Характеристика відкритого успадкування. Ознаки захищеного успадкування. Особливості закритого успадкування.

Тема 7. Множинне успадкування. Механізми успадкування декількох базових класів.

Алгоритм викликів конструкторів при успадкуванні класів. Прикладні аспекти викликів деструкторів при успадкуванні класів. Можливості передачі параметрів у конструктори, якщо клас успадкований від кількох класів. Недоліки множинного успадкування.

Тема 8. Оголошення класів у заголовочних файлах.

Зміст заголовкових файлів із розширенням *.h? Механізм під'єднання файлу із розширенням *.h до проекту. Використання стандартних файлів C++ розширення *.h. Причини винесення оголошення класів у окремі файли. Переваги оголошення класів у окремих файлах. Етап приєднання середовищем програмування заголовкових файлів.

**4.СТРУКТУРА ЗАЛІКОВОГО КРЕДИТУ З ДИСЦИПЛІНИ
«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»**

Назва теми	Кількість годин					
	Лекції	Практ. заняття	Сам. робота	Інд. робота	Тренінг	Контр. заходи
Змістовий модуль I. Основні парадигми об'єктно-орієнтованого програмування						
Тема 1. Вступ до об'єктно-орієнтованого програмування. Поняття класу та об'єкту.	4	4	21	-	4	Поточне опитування
Тема 2. Конструктори та деструктори класу.	4	4	10	1		
Тема 3. Робота із масивами об'єктів.	4	4	10	-		
Тема 4. Види класів. Вкладені класи	4	2	10			
Тема 5. Перевантаження операторів.	2	4	10			
Змістовий модуль II. Реалізація принципів ООП мовою C++						
Тема 6. Успадкування. Одинарне успадкування.	4	4	10	1	4	Поточне опитування
Тема 7. Множинне успадкування. Механізми успадкування декількох базових класів.	4	4	10	1		
Тема 8. Оголошення класів у заголовочних файлах.	4	4	10	1		
Всього:	30	30	78	4	8	

5. ТЕМАТИКА ПРАКТИЧНИХ ЗАНЯТЬ

Практичне заняття № 1

Тема: Вступ до об'єктно-орієнтованого програмування. Програмування класів
Мета: Навчитись проектувати та оголошувати класи, створювати об'єкти класів

Завдання для практичного виконання

Варіант 1. Описати клас Square, з властивістю довжина сторони квадрата. Методи: set- та get-, обчислення площі та периметра. Написати програму, яка демонструватиме роботу з цим класом.

Варіант 2. Описати клас Circle, з властивістю радіус. Методи: set- та get-, обчислення площі круга та довжини кола. Написати програму, яка демонструватиме роботу з цим класом.

Варіант 3. Описати клас Date з властивостями день, місяць, рік. Методи: set- та get-, попередній день, наступний день. Написати програму, яка демонструватиме роботу з цим класом.

Практичне заняття № 2

Тема: Робота з класами. Методи класу

Мета: Навчитись проектувати та оголошувати власні класи, а також модифікувати уже розроблені

Завдання для практичного виконання

Варіант 1. Створіть для класу ZooAnimal конструктор, що створюватиме за замовчуванням об'єкт із наступними параметрами з параметрами: "Nameless" для name, 9999 для whichCage, 101 (January 1) для weighDay, та 100 для pounds.

Варіант 2. Для заданого класу напишіть функцію int daysSinceLastWeighed (int today), яка рідрахеє, скільки днів минуло від останнього зважування. Параметром функції є сьогоднішня дата. Результатом даного завдання є програма, що демонструє роботу із класом. Для деяких методів не вказано, як описувати реалізацію, тому для них залишаєте тільки оголошення.

Практичне заняття № 3

Тема: Методи класу. Конструктор та деструктор класу

Мета: Навчитись програмувати дані-методи класу, викликати методи, програмувати та викликати конструктори різних типів для класу, програмувати та викликати деструктори класу.

Завдання для практичного виконання

Для кожного завдання розробити клас, створити конструктор ініціалізації, конструктор копіювання та перетворення типів, set-, get-методи. Розробити програму, яка демонструватиме роботу із даним класом.

Варіант 1. Клас RightTrapezoid із наступними атрибутами: висота, бічна сторона та менша основа. Визначити для даного класу функції, що будуть визначати площу та периметр трапеції.

Варіант 2. Клас Purchase із наступними атрибутами: ціна, термін придатності (до якої дати товар придатний для використання). Визначити для цього класу функції, які повертають вартість покупки із знижкою у 15% та перевіряють, чи товар придатний для використання, якщо зараз 22.08.2024.

Практичне заняття № 4

Тема: Програмування задач, що використовують масиви об'єктів

Мета: Навчитись використовувати масиви об'єктів при розв'язуванні задач, ініціалізувати такі масиви, викликати методи для елементів масивів

Завдання для практичного виконання

Варіант 1.

1. Створити клас Storage, який містить наступні поля:

- Name – назва товару;
- Type – одиниця вимірювання;
- Quantity – кількість одиниць товару;
- Cost – ціна одиниці товару.

2. Написати програму, що використовує даний клас і виконує наступні дії:

- вводить з клавіатури масив даних Mall що складається з N змінних типу Storage;
- Виводить на екран ціну та кількість товару, назва якого вводиться з клавіатури, або виводить повідомлення про його відсутність.

Варіант 2.

1. Створити клас Bus, який містить наступні поля:

- Name – назва пункту призначення;
- Numer – номер рейсу;
- Date – дата відправлення;
- Time – час відправлення.

2. Написати програму, що використовує даний клас і виконує наступні дії:

- вводить з клавіатури масив даних BusStation що складається з n змінних типу Bus;
- виводить на екран всі рейси, які відправляються після 17.00 у задану дату.

Практичне заняття № 5

Тема: Програмування задач на використання композицій класів

Мета: Навчитись використовувати контейнерні класи про розробці програм

Завдання для практичного виконання

В усіх завданнях розробити конструктор для об'єктів класу, деструктор, set-, get- методи та програму, що буде демонструвати роботу із створеним класом.

Варіант 1. Розробити клас Computer, який містить об'єкти класів Mouse, Monitor, Keyboard, SystemUnit. Визначити необхідні елементи даних. Створити масив об'єктів класу Computer та визначити комп'ютер із мінімальним відхиленням параметрів від заданих.

Варіант 2. Розробити клас Car, який містить об'єкт класу Engine. У класі Engine визначити дані про об'єм двигуна, потужність та заводський номер. Клас Car додатково містить номер державної реєстрації, марку та колір. Визначити методи для зміни марки, номера та заводського номера двигуна для класу Car.

Практичне заняття № 6

Тема: Програмування задач, що використовують перевантаження операторів

Мета: Навчитись здійснювати перевантаження бінарних та унарних операторів, використовувати нове значення оператора у програмах

Завдання для практичного виконання

В усіх завданнях розробити конструктор для об'єктів класу, деструктор, set-, get- методи та програму, що буде демонструвати роботу із створеним класом та дію операторів, для яких було змінено значення.

Варіант 1. Створити клас Fraction, що є відношенням двох цілих чисел. Перевантажити операції додавання, віднімання, множення та ділення.

Варіант 2. Створити клас RealNumber (у закритій частині класу знаходиться дане дійсного типу). Перевантажити оператори () – повертає дробову частину числа, [] – повертає цілу частину числа.

Практичне заняття № 7

Тема: Програмування задач із застосуванням одинарного успадкування

Мета: Навчитись реалізовувати одинарне успадкування у класах, використовувати цей принцип об'єктно-орієнтованого програмування у прикладних програмах

Завдання для практичного виконання

Варіант 1. Створити класи транспортних засобів: Auto, Truck, Boat і Plane. Створити з них ієрархії. В основу ієрархії покласти клас Vehicle зі спільними для усіх цих класів елементами. Визначити функції виведення, конструктори і деструктори. Розробити програму, що демонструватиме роботу з класом.

Варіант 2. Використовуючи ієрархію та успадкування, створити класи Window, WindowWithTitle і WindowWithButton. Визначити необхідні конструктори, деструктори та метод зміни назви заголовку вікна. Зімітувати натискання кнопки для закривання вікна. Розробити програму, що демонструватиме роботу з класом.

Практичне заняття № 8

Тема: Програмування задач із застосуванням множинного успадкування

Мета: Навчитись реалізовувати множинне успадкування у класах, використовувати цей принцип об'єктно-орієнтованого програмування у прикладних програмах.

Завдання для практичного виконання

Варіант 1. Описати класи Жанр, Носій та Фільм. Клас Фільм є нащадком класів Жанр та Носій. Клас Жанр має поля «назва жанру», «характерні особливості» та методи, створення та виведення жанру. Клас Носій має поля «тип носія», «кольорова гама» та «тривалість зберігання» та методи створення та виведення. Клас Фільм має власні поля «назва», «автор сценарію», «режисер», «тривалість» та методи створення, виведення усіх характеристик фільму. З використанням цих класів скласти програму опису деякого фільму та вивести інформацію щодо нього.

Варіант 2. Описати класи ХудожнійТвір, Техніка та Картина. Клас Картина є нащадком класів ХудожнійТвір та Техніка. Клас ХудожнійТвір має поля «назва твору», «рік написання», «автор», «жанр» та методи, створення, виведення художнього твору. Клас Техніка описує використану техніку живопису. Цей клас має поля «назва техніки» та «матеріал» та методи створення та виведення. Клас Картина має власні поля «ширина», «висота», «вартість» та методи створення та виведення усіх характеристик картини, а також методи читання та зміни вартості. З використанням цих класів скласти програму опису деякої картини та вивести інформацію щодо неї.

7. Тренінг з дисципліни.

Порядок проведення тренінгу:

Вступна частина проводиться з метою ознайомлення студентів з темою тренінгу.

Організаційна частина полягає у створенні робочого настрою у колективі студентів.

Практична частина реалізується шляхом виконання завдань з певних проблемних питань теми тренінгу.

Підведення підсумків.

Обговорення результатів виконаних завдань.

Обмін думками з питань, що виносились на тренінг. Рекомендується проведення тренінгу за наступною темою:

1. Принципи об'єктно-орієнтованого програмування.
2. Типові задачі об'єктно-орієнтованого програмування.

8. Самостійна робота

Для успішного вивчення і засвоєння дисципліни «Об'єктно-орієнтоване програмування» студенти повинні володіти значним обсягом інформації, частину якої вони отримують і опрацьовують шляхом самостійної роботи.

- 1 Об'єктний підхід в об'єктно-орієнтованому проектуванні.

- 2 Класифікація в об'єктно-орієнтованому аналізі.
- 3 Об'єктно-орієнтоване проектування. Система позначень.
- 4 Діаграми класів.
- 5 Діаграми переходів станів.
- 6 Діаграми об'єктів.
- 7 Часові діаграми.
- 8 Модульні діаграми.
- 9 Діаграми процесів.
- 10 Ідентифікація зв'язків між класами та об'єктами.
- 11 Об'єктно-орієнтоване програмування в життєвому циклі розробки програми.
- 12 Переваги та недоліки об'єктно-орієнтованого програмування.
- 13 Відношення між об'єктами.
- 14 Відношення між класами.
- 15 Взаємозв'язки класів та об'єктів.
- 16 Перевантаження операторів.
- 17 Наслідування.
- 18 Віртуальні функції.
- 19 Абстрактні класи.
- 20 Контроль доступу до членів класу.
- 21 Шаблони класів.
- 22 Параметризовані функції.
- 23 Обробка виняткових ситуацій.
- 24 Множинне наслідування. Керування доступом.
- 25 Інформація про тип на етапі виконання.
- 26 Стандартні контейнери.
- 27 Визначення нового контейнера.
- 28 Алгоритми стандартної бібліотеки.
- 29 Ітератори і розподіл пам'яті.
- 30 Потоки вводу/виводу.
- 31 Стандартні бібліотеки чисельних методів.
- 32 Процес розробки програмного забезпечення, його мета, головні етапи.
- 33 Цілі програмування та розробки програмного забезпечення.
- 34 Використання змінних.
- 35 Загальноживані оператори.
- 36 Умовні конструкції.
- 37 Цикли. Типи циклів.
- 38 Процедури.
- 39 Обробка помилок.
- 40 Скалярні стандартні типи даних в мові C#.
- 41 Скалярні типи користувача в мові C#.
- 42 Основні операції в мові C#.
- 43 Загальна структура програми мовою C#.
- 44 Умовні оператори мови C#.
- 45 Оператори циклу у мові C#.
- 46 Введення і виведення інформації у мові C#.
- 47 Методи в мові C#.
- 48 Рекурсія. Пряма та непряма рекурсія.
- 49 Типи параметрів у мові C#.
- 50 Масиви у мові C#.
- 51 Основні принципи об'єктно-орієнтованого підходу в програмуванні.
- 52 Поняття інкапсуляції. Організація взаємодії об'єктів.
- 53 Типи-значення та типи-посилання.
- 54 Способи створення об'єктів.

9. Методи навчання

У навчальному процесі застосовуються: лекції, практичні заняття, консультації, самостійна робота, робота у групах, метод опитування, тестування, реферування, есе, підготовка і презентація результатів досліджень.

10. Засоби оцінювання та методи демонстрування результатів навчання

В процесі вивчення дисципліни “Об’єктно-орієнтоване програмування” використовуються наступні методи оцінювання навчальної роботи студента:

- поточне опитування;
- стандартизовані тести;
- залікове модульне тестування та опитування;
- презентації результатів виконання завдань;
- ректорська контрольна робота;
- тренінг;
- **екзамен.**

11. Політика оцінювання

Політика щодо дедлайнів і перескладання. Для виконання індивідуальних завдань і проведення контрольних заходів встановлюються конкретні терміни. Перескладання модулів відбувається з дозволу дирекції факультету за наявності поважних причин (наприклад, лікарняний).

Політика щодо академічної доброчесності. Використання друкованих і електронних джерел інформації під час контрольних заходів та екзаменів заборонено.

Політика щодо відвідування. Відвідування занять є обов’язковим компонентом оцінювання. За об’єктивних причин (наприклад, карантин, воєнний стан, хвороба, закордонне стажування) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу з дозволу дирекції факультету.

12. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Об’єктно-орієнтоване програмування» визначається як середньозважена величина, залежно від питомої ваги кожної складової залікового кредиту:

Модуль 1		Модуль 2		Модуль 3	Модуль 4	Модуль 5
10%	10%	10%	10%	5%	15%	40%
Поточне опитування	Модульний контроль 1	Поточне опитування	Модульний контроль 2	Тренінг	Самостійна робота	Екзамен
Середнє арифметичне від отриманих оцінок під час занять по темах 1-5 не рідше одного разу на 2 практичні заняття	Контрольна робота	Середнє арифметичне від отриманих оцінок під час занять по темах 5-8 не рідше одного разу на 2 практичні заняття	Контрольна робота	Середнє арифметичне від отриманих оцінок під час виконання завдань тренінгу	Оцінка за виконання аналітично-практичного завдання	Теоретичне питання – 20 балів, 10 тестів – 50 балів, задача 30 балів

Шкала оцінювання:

За шкалою ЗУНУ	За національно	За шкалою ECTS
90-100	«відмінно»	A (відмінно)
85-89	«добре»	B (дуже добре)
75-84		C (добре)
65-74	«задовільно»	D (задовільно)
60-64		E (достатньо)
35-59	«незадовільно»	FX (незадовільно з можливістю повторного
1-34		F (незадовільно з обов'язковим повторним курсом)

13. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1.	Технічне забезпечення: мультимедійний проєктор, ноутбук, проєкційний екран. Опорний конспект лекцій (електронний варіант)	1-8
2.	Завдання для практичних занять під керівництвом викладача та для самостійної роботи (електронний варіант)	
3.	Індивідуальні завдання для самостійного виконання (електронний варіант)	1-8
4.	Базове програмне забезпечення: ОС Windows. Стандартне програмне забезпечення базових інформаційних технологій: MS Office (Word, Excel, PowerPoint, Microsoft Visio). Телекомунікаційне програмне забезпечення (Internet Explorer, Google Chrome, Viber тощо)	1-8
5.	Комунікаційна навчальна платформа (Moodle) для організації дистанційного навчання (за необхідності).	1-8

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Богач І. В., Довгалець С. М., Дубовой В. М. Алгоритми розв'язання задач з програмування. Розв'язник. Вінниця: ВНТУ, 2017. 119 с.
2. Бойко Б. І., Омельчук Л. Л., Русіна Н. Г. Об'єктно-орієнтоване програмування. Лабораторний практикум: навч. посіб. Київ : КНУ ім. Тараса Шевченка, 2016. 90 с.
3. Бублик В. В. Об'єктно-орієнтоване програмування. Київ : ІТ-книга, 2015. 640 с.
4. Вакалюк Т. А., Шевчук Л. Д., Постова С. А. Структурне та візуальне програмування Навчальний посібник для студентів фізико-математичного факультету. Переяслав-Хмельницький : вид-во ПХДПУ, 2019. 318 с.

6. Грицюк Ю., Рак Т. Програмування мовою С++. Львів: Вид-во ЛДУ БЖД, 2011. 290 с.
7. Грицюк Ю., Рак Т. Об'єктно-орієнтоване програмування мовою С++. Львів: Вид-во ЛДУ БЖД, 2011. 403 с.
8. Жуковський С. С., Вакалюк Т. А. Об'єктно-орієнтоване програмування мовою С++. Навчально-методичний посібник для студентів напряму 6.040302 "Інформатика". Житомир: Вид-во ЖДУ, 2016. 100 с.
9. Зубенко В. В., Омельчук Л. Л. Програмування. Поглиблений курс. Ктїв : ВПЦ "Київський університет", 2011. 624 с.
10. Каплун В. А., Баришев Ю. В., Остапенко А. В. Технологія програмування. Лабораторний практикум: навч. посіб. Вінниця : ВНТУ, 2015. 125 с.
11. Коротеєєва Т. О. Алгоритми та структури даних: навч. посіб. Львів : Видавництво Львівської політехніки, 2014. 280 с. 120
12. Кравець О. П. Об'єктно-орієнтоване програмування: навч. посіб. Львів: Видавництво Львівської політехніки, 2012. 626 с.
13. Крєневич А. П., Обвінцев О. В. С++ у задачах і прикладах : навчальний посібник із дисципліни "Інфор-матика та програмування". К : Видав-ничополіграфічний центр "Київський університет", 2011. 208 с.
14. Нікітченко М. С. Теорія програмування. Частина 1. Навчальний посібник Ніжин: Видавництво НДУ імені Миколи Гоголя, 2010. 121 с.
15. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування. Частина 1: навчальний посібник. Львів : Видавництво «Новий Світ-2000», 2020. 320 с.
16. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування. Частина 2: навчальний посібник. Львів : Видавництво «Новий Світ-2000», 2020. 337 с.
17. С++. Основи програмування. Теорія та практика: підручник / О. Г. Трофименко та ін. Одеса: Фенікс, 2010. 544 с.
18. С++. Теорія та практика: навч. посіб.з грифом МОНУ/ О. Г. Трофименко та ін. Одеса : ВЦ ОНАЗ, 2011. 587 с.
19. Татарчук Д. Д., Діденко Ю. В. Програмування мовами С та С++: навч. посіб. Ктїв : НТУУ" КПІ", 2012. 112 с.
20. Трофименко О. Г., Буката Л. М., Лєонов Ю. Г. Алгоритмізація обчислювальних процесів і особливості програмування в С++: метод. посіб. Одеса : ЩОНАЗ, 2009. 93 с.
21. Трофименко О. Г., Прокоп Ю. В., Логінова Н. І., Задерейко О. В. С++. Алгоритмізація та програмування: підручник. Одеса : Фенікс, 2019. 477 с.
22. Шевчук І.Б. Конспект лекцій з навчальної дисципліни "Алгоритмізація та програмування". Львів : Львівський національний університет ім. Івана Франка, 2018. 30с.