

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ

Декан

Савенко О.С.

1

вересня

2022 р.

## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### Об'єктно-орієнтоване програмування

Назва

**Галузь знань** 12 – Інформаційні технології

**Спеціальність** 123 – Комп'ютерна інженерія очна денна форма здобуття освіти

**Освітня програма** Комп'ютерна інженерія та програмування

**Статус дисципліни:** обов'язкова, дисципліна професійної підготовки

**Факультет** – інформаційних технологій

**Кафедра** – Комп'ютерної інженерії та інформаційних систем

Форма здобуття освіти	Курс	Семестр	Загальне навантаження		Кількість годин							Форма семестрового контролю		
			Кредити ЄКТС	Години	Аудиторні заняття					Курсовий проєкт	Курсова робота	Залік	Іспит	
					Разом	Лекції	Лабораторні роботи	Практичні заняття	Індивідуальна робота студента					Самостійна робота, в т.ч. ІРС
ОД	2	3	7.0	210	102	34	34	34		108	-	-		
<b>Разом</b>			7.0	210	102	34	34	34		108	-	-		1

Робоча програма складена на основі стандарту вищої освіти зі спеціальності 123 Комп'ютерна інженерія, освітньо-професійної програми та навчального плану

Програма складена

Підпис

Лисенком С.М.

Ініціали, прізвище викладача(ів)

Схвалена на засіданні кафедри Комп'ютерної інженерії та інформаційних систем

Протокол 1 від 12.08.2022 р.

Зав. кафедри КІІС

Підпис

Говорухенко Г.О.

Ініціали, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій

Голова Вченої ради

Підпис

Савенко О.С.

Ініціали, прізвище

Хмельницький 2022

## ВСТУП

**Мета викладання дисципліни.** Дисципліна “Об’єктно-орієнтоване програмування” відноситься до циклу дисциплін професійної підготовки, забезпечує базову підготовку студентів спеціальності “Комп’ютерна інженерія” з програмування та характеризується широким міждисциплінарним підходом.

Метою курсу є оволодіння студентами основними принципами і методами об’єктно-орієнтованого програмування, які в подальшому можуть сприяти їх успішному застосуванню в професійній діяльності.

Таким чином, мета і завдання дисципліни “Об’єктно-орієнтоване програмування” навчити застосовувати принципи об’єктно-орієнтованого програмування на всіх етапах життєвого циклу прикладної програмної системи, починаючи з аналізу вимог до програмної системи і її попереднього проектування, і закінчуючи її реалізацією, тестуванням і наступним супроводом.

**Предмет дисципліни.** Основи програмування мовою C++ на прикладі компілятора Microsoft ©Visual Studio©.

**Завдання дисципліни.** Навчити застосовувати принципи об’єктно-орієнтованого програмування на прикладі компілятора Microsoft ©Visual Studio©; розглянути всі основні поняття C++, такі як об’єкти, класи, наслідування, перевантаження, поліморфізм, опрацювання виняткових ситуацій, потоки; розглянути такі специфічні питання, як мета-класи, делегування, шаблони; навчити об’єктно-орієнтованому підходу до розробки і реалізації прикладних програмних систем; довести доцільність і плідотворність систематичного застосування об’єктно-орієнтованого підходу на всіх етапах життєвого циклу прикладної програмної системи.

Після вивчення дисципліни "**Об’єктно-орієнтоване програмування**" студент має досягти таких результатів навчання (сукупність знань, умінь, навичок, компетентностей):

### **знати:**

- об’єкт, предмет, задачі, проблематику дисципліни та її основні розділи;
- наукові і математичні положення, що лежать в основі функціонування програмних систем; базові поняття й визначення, використовувані у галузі програмної інженерії; основи проектування програмного забезпечення мовою C++;
- інновації у галузі програмної інженерії

### **уміти:**

- застосовувати основи комп’ютерної інженерії для розв’язування задач синтезу програмних систем на основі об’єктно-орієнтованого підходу із застосуванням мови програмування C++;
- системно мислити та застосовувати творчі здібності до формування принципово нових ідей щодо розроблення програмних систем засобами ООП мовою програмування мовою C++;
- застосовувати знання принципів об’єктно-орієнтованого програмування мовою C++ для ідентифікації, формулювання і розв’язування технічних задач спеціальності;
- вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем застосовуючи об’єктно-орієнтований підхід до програмування мовою C++;
- застосовувати теорію та методи системного аналізу, математичного і комп’ютерного моделювання при проектуванні та розробленні програмного забезпечення та програмних систем на основі об’єктно-орієнтованого підходу мовою C++;
- оцінювати результати роботи, захищати, пояснювати та аргументувати розроблену програмну систему мовою C++, отримані результати;
- вміти здійснювати пошук інформації в різних джерелах для розв’язання задач побудови ПЗ на основі об’єктно-орієнтованого підходу програмування мовою C++;

### **бути здатним:**

- розв’язувати складні задачі і проблеми в галузі проектування програмного забезпечення мовою C++, що передбачає проведення досліджень та/або здійснення інновацій;
- абстрактно мислити, аналізувати і синтезувати програмні системи на основі об’єктно-орієнтованого підходу програмування мовою C++;

- застосовувати знання принципів об'єктно-орієнтованого програмування на практиці;
- розробляти програмне забезпечення, програмні компоненти комп'ютерних систем та мереж, Інтернет додатки, кіберфізичних систем з використанням об'єктно-орієнтованих методів програмування мовою С++;
- створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж із застосуванням об'єктно-орієнтованого підходу програмування мовою С++;
- проектувати програмні системи та їхні компоненти із застосуванням об'єктно-орієнтованих методів програмування мовою С++ з урахуванням усіх аспектів їх життєвого циклу ПЗ, включаючи створення, налаштування, експлуатацію та обслуговування.

**Компетентності, на формування яких спрямовано ОК:**

Інтегральна – Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп'ютерній галузі або навчання, що передбачає застосування теорій та методів комп'ютерної інженерії і характеризується комплексністю та невизначеністю умов

ЗК2. Здатність вчитися і оволодівати сучасними знаннями

ЗК3. Здатність застосовувати знання у практичних ситуаціях

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово

ЗК7. Вміння виявляти, ставити та вирішувати проблеми

ЗК11. Здатність до розуміння предметної галузі та професійної діяльності

ЗК12. Здатність використовувати інформаційні та комунікаційні технології

ЗК13. Здатність розв'язувати поставлені задачі та приймати відповідні рішення

ФК1. Здатність застосовувати законодавчу та нормативно-правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії.

ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

ФК15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати та захищати прийняті рішення.

ФК17. Здатність забезпечувати проектування та розроблення якісних програмних і технічних засобів комп'ютерних систем та мереж

ФК18. Здатність розробляти бізнес-рішення та оцінювати нові технологічні пропозиції

**Програмні результати навчання, на забезпечення яких спрямовано ОК:**

ПРН3 – Знати новітні технології в галузі комп'ютерної інженерії

ПРН5. Мати знання основ економіки та управління проектами.

ПРН6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПРН8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.

ПРН11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

ПРН12. Вміти ефективно працювати як індивідуально, так і у складі команди.

ПРН15. Вміти виконувати експериментальні дослідження за професійною тематикою.

ПРН16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

ПРН17. Спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською).

ПРН18. Використовувати інформаційні технології для ефективного спілкування на

професійному та соціальному рівнях.

ПРН19. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

ПРН20. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

ПРН21. Якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики.

ПРН23. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання програмно-технічних засобів комп'ютерних систем та мереж.

## ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	3
Кількість встановлених кредитів ЄКТС	7
Форми здобуття освіти	Очна денна

**Результати навчання. ЯК ПРН** Студент, який успішно завершив вивчення дисципліни, повинен: знати новітні технології в галузі комп'ютерної інженерії; мати знання основ економіки та управління проектами; вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей; вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності; вміти системно мислити та застосовувати творчі здібності до формування нових ідей; вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання; вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії; вміти ефективно працювати як індивідуально, так і у складі команди; вміти виконувати експериментальні дослідження за професійною тематикою; вміти оцінювати отримані результати та аргументовано захищати прийняті рішення; спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською); використовувати інформаційні технології для ефективного спілкування на професійному та соціальному рівнях; мати здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення; усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення; якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики; використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання програмно-технічних засобів комп'ютерних систем та мереж.

**Зміст навчальної дисципліни.** Особливості мови програмування C++. Класи. Відношення між класами. Наслідування. Поліморфізм. Перевантаження операторів. Шаблони в C++. Бібліотека STL. Контейнери, ітератори, предикати в C++. Обробка подій. Виключні ситуації. Windows Form Applications.

**Запланована навчальна діяльність:** лекції – 34 год., лабораторні заняття – 34 год., практичні заняття – 34 год. самостійна робота – 108 год.; разом – 210 год.

**Методи навчання:** словесні, наочні, проблемно-пошукові (лекції); пояснювально-ілюстративні, практичні, частково-пошукові (практичні та лабораторні заняття), дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

**Форми оцінювання результатів навчання:** захист лабораторних та практичних робіт, тестовий контроль, підсумковий контрольний захід.

**Форма семестрового контролю:** іспит

### Навчальні ресурси:

1. Олексій Васильєв. Програмування C++ в прикладах і задачах. Ліра-К. 2017. с. 382
2. Галкін О.В., Верес М.М. Мова програмування C++. ДП «Видавничий дім «Персонал» .2017. с.260
3. В.В. бублик. Об'єктно-орієнтоване програмування. Київ. ІТ-книга. 2015. с.640
4. В.А. Данілова. Об'єктно-орієнтоване програмування. ПРАКТИКУМ. Київ КПІ ім. Ігоря Сікорського 2021. с.121
5. Б'ярне Страуструп.Експерсія по C++ (2-е видання) (C++ In-Depth Series) 2-е видання. Addison-Wesley Professional. 2018. С.256 (англ)
6. Курт Гюртеон. Оптимізований C++ (Optimized C++). O'Reilly Media. 2016. С.338 (англ)
7. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua>.
8. Електронна бібліотека університету. Доступ до ресурсу: [http://lib.khnu.km.ua/asp/php\\_f/page\\_lib.php](http://lib.khnu.km.ua/asp/php_f/page_lib.php).

**Викладач:** доктор технічних наук, професор Лисенко С.М.

## 1. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин відведених на:			
	Лекції	Лабораторні роботи	Практичні роботи	СРС
<b>Тема 1.</b> Особливості мови програмування C++. Робота з файлами	4	4	4	12
<b>Тема 2.</b> Класи.	8	8	8	24
<b>Тема 3.</b> Наслідування.	4	4	4	12
<b>Тема 4.</b> Поліморфізм.	4	4	4	12
<b>Тема 5.</b> Перевантаження операторів.	2	2	2	6
<b>Тема 6.</b> Шаблони в C++.	2	2	2	7
<b>Тема 7.</b> Бібліотека STL. Контейнери, ітератори, предикати в C++	4	4	4	14
<b>Тема 8.</b> Обробка подій. Об'єктно-орієнтована програма, яка керована подіями.	2	2	2	7
<b>Тема 9.</b> Виключні ситуації.	2	2	2	7
<b>Тема 10.</b> Windows Form Applications.	2	2	2	7
<b>Години</b>	34	34	34	108
<b>Разом</b>	<b>210 (7.0 кредитів)</b>			

## 2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### 2.1. Зміст лекційного курсу\*

№ п/п	Перелік тем лекцій, їх анотація	Години
1	<b>Тема 1. Особливості мови програмування C++. Робота з файлами</b> <b>Лекція№1.</b> Локальні та глобальні змінні. Підпрограми та їх аргументи. Визначення даних. Модифікатор const. Модифікатор volatile Перевантаження функцій. [1, 3, 4].	2
2	<b>Тема 1. Особливості мови програмування C++. Робота з файлами</b> <b>Лекція№2.</b> Потоки. Робота з файлами [2, 3, 6].	2
3	<b>Тема 2. Класи в C++.</b> <b>Лекція№3.</b> Поняття класу в C++. Поняття об'єкту та об'єктно-орієнтоване програмування. Визначення методів класу поза класом. Методи класу. Модифікатори доступу. Використання оператора глобального дозволу для елементів класу. [1, 5, 7]	2
4	<b>Тема 2. Класи в C++.</b> <b>Лекція№4.</b> Динамічний розподіл пам'яті. Ініціалізація виділеної пам'яті. Виділення пам'яті для масивів. Виділення пам'яті для об'єктів класів. [1, 8].	2
5	<b>Тема 2. Класи в C++.</b> <b>Лекція№5.</b> Конструктори. Конструктори за замовчуванням. Конструктори з параметрами. Приватні конструктори. Конструктори копіювання. Деструктори. Загальнодоступні деструктори. Приватні деструктори. Статичні функції та елементи даних. Дружні функції та класи. Ключове слово this [1, 3, 4, 8]	2
6	<b>Тема 2. Класи в C++.</b> <b>Лекція№6.</b> Відношення між класами. Узагальнення. Асоціація. Агрегація. Композиція [5, 6, 9]	2
7	<b>Тема 3. Наслідування.</b> <b>Лекція№7.</b> Поняття наслідування. Просте наслідування. Наслідування і захищені члени. Управління доступом до членів базового класу. Розв'язання конфлікту імен. Керування доступом до членів базового класу. Захищене наслідування. Конструктори, деструктори і наслідування [1, 6, 10, 11].	2
8	<b>Тема 3. Наслідування.</b> <b>Лекція№8.</b> Наслідування. Передача параметрів конструктору базового класу. Надання доступу при наслідуванні. Множинне наслідування. Віртуальні класи при наслідуванні. [2,6,7,10].	2
9	<b>Тема 4. Поліморфізм.</b> <b>Лекція№9.</b> Віртуальні функціїта невіртуальн функції. Виклик віртуальної функції за допомогою посилання на об'єкт базового класу. Наслідування атрибуту virtual. поліморфізм. [2,6,7,11].	2
10	<b>Тема 4. Поліморфізм</b> <b>Лекція№10.</b> Абстрактні класи. Застосування віртуальних функцій. Чисто віртуальні функції. Віртуальні деструктори. Віртуальні конструктори. [2,6,7,10].	2
11	<b>Тема 5. Перевантаження операторів</b> <b>Лекція№11.</b> Створення операторної функції-члена. Перевантаження скорочених операторів присвоювання. Обмеження на перевантажені оператори. Створення префіксної і постфіксної форм операторів інкрементації і декрементации. Перевантаження операторів за допомогою дружніх функцій. Застосування дружніх функцій для перевантаження операторів "++" і "--".	2

	Перевантаження операторів new і delete. Перевантаження деяких спеціальних операторів. [1,4,7,10].	
12	<b>Тема 6. Шаблони в С++.</b> <b>Лекція№12.</b> Шаблони функцій. Шаблони класів. Оголошення об'єктів, що базуються на шаблоні класу. [2,6,7,10].	2
13	<b>Тема 7. Стандартна бібліотека шаблонів STL</b> <b>Лекція№13.</b> Стандартна бібліотека шаблонів. Складові STL. Класи-контейнери. [1,9,12].	2
14	<b>Тема 7. Стандартна бібліотека шаблонів STL.</b> <b>Лекція№14.</b> Асоціативні контейнери. Алгоритми. Предикати. Застосування функторів. [1,9,12].	2
15	<b>Тема 8. Обробка подій. Windows Form Applications</b> <b>Лекція№15.</b> Створення події. Обробка подій в С++. [6,7,13].	2
16	<b>Тема 9. Обробка подій. Windows Form Applications</b> <b>Лекція№16.</b> Компоненти, що утворюють інтерфейс між користувачем і додатком. Простір імен System. Основні компоненти. Властивості компонентів. [3, 9, 14].	2
17	<b>Тема 9. Виключні ситуації.</b> <b>Лекція№17.</b> Основи обробки виняткових ситуацій. Перехоплення класів виняткових ситуацій. Обробка похідних виняткових ситуацій. Обмеження виключних ситуацій. Функції terminate() і unexpected(). [1, 8, 15]	2
	<b>Загалом:</b>	<b>34</b>



## 2.2 Зміст лабораторних занять

№ п/п	Теми лабораторних робіт	Кількість годин
1	<i>Лабораторна робота №1.</i> Основні відмінності мови програмування C++ від Cі [1,4].	4
2	<i>Лабораторна робота №2.</i> Класи, методи, оператори new, delete в C++[1-4].	4
3	<i>Лабораторна робота №3.</i> Конструктори та деструктори, статичні функції та елементи даних, дружні функції та класи, конструктор копіювання в C++ [2,6,7,10].	4
4	<i>Лабораторна робота №4.</i> Наслідування, множинне наслідування [2,6,7,10].	4
5	<i>Лабораторна робота № 5.</i> Поліморфізм, віртуальні функції, абстрактні класи та виняткові ситуації. [1-4,10].	4
6	<i>Лабораторна робота № 6.</i> Перевантаження операторів, шаблони в C++ [1,9,10].	4
7	<i>Лабораторна робота №7.</i> Бібліотека STL. Контейнери, літератори, предикати в C++. [1,2,6,7].	4
8	<i>Лабораторна робота №8.</i> Windows Form Applications. [6,9,10]	4
9	<i>Підсумкове заняття.</i>	2
	<b>Загалом:</b>	<b>34</b>

### 2.3 Зміст практичних занять

№ п/п	Теми практичних робіт	Кількість годин
1	<i>Практична робота №1.</i> Основні відмінності мови програмування C++ від Cі [1,4].	4
2	<i>Практична робота №2.</i> Класи, методи, оператори new, delete в C++[1-4].	4
3	<i>Практична робота №3.</i> Конструктори та деструктори, статичні функції та елементи даних, дружні функції та класи, конструктор копіювання в C++ [2,6,7,10].	4
4	<i>Практична робота №4.</i> Наслідування, множинне наслідування [2,6,7,10].	4
5	<i>Практична робота № 5.</i> Поліморфізм, віртуальні функції, абстрактні класи та виняткові ситуації. [1-4,10].	4
6	<i>Практична робота № 6.</i> Перевантаження операторів, шаблони в C++ [1,9,10].	4
7	<i>Практична робота №7.</i> Бібліотека STL. Контейнери, літератори, предикати в C++. [1,2,6,7].	4
8	<i>Практична робота №8.</i> Windows Form Applications [6,9,10]	4
9	<i>Підсумкове заняття.</i>	2
	<i>Загалом:</i>	34

## 2.4 Зміст самостійної (індивідуальної) роботи

Самостійна робота студентів денної форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту практичних та лабораторних робіт, тестування з теоретичного матеріалу, виконанні індивідуальних завдань.

№ п/п	Теми самостійної роботи	Кількість годин
1	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №1 та до практичної роботи №1.	6
2	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №2.	6
3	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №1. Підготовка до захисту практичної роботи №1.	6
4	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №3.	6
5	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №2. Підготовка до захисту практичної роботи №2.	6
6	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №4.	6
7	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №3. Підготовка до захисту практичної роботи №3.	6
8	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №5.	6
9	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №4. Підготовка до захисту практичної роботи №4.	6
10	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №6. Підготовка до тестування за темами 1-5.	6
11	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №5. Підготовка до захисту практичної роботи №5.	6
12	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №7.	7
13	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №6. Підготовка до захисту практичної роботи №6.	7
14	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №8.	7
15	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №7. Підготовка до захисту практичної роботи №7.	7
16	Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи №8. Підготовка до практичної роботи №8. Підготовка до тестування за темами 6-10.	7
17	Опрацювання лекційного матеріалу. Підготовка до підсумкових лабораторного та практичного занять	7
<b>Загалом:</b>		<b>108</b>

Керівництво самостійною роботою та контроль за виконанням індивідуального завдання здійснює викладач згідно з розкладом консультацій у позаурочний час.

## 3. МЕТОДИ НАВЧАННЯ

Лекції проводяться, в основному, з використанням словесних, наочних, проблемно-пошукових методів; практичні та лабораторні заняття проводяться пояснювально-ілюстративними методами, практичними та частково-пошуковими методами; самостійна робота передбачає виконання індивідуальних завдань із залученням практичних, дослідницьких, частково-пошукових методів.

## 4. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних, практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни.

Семестровий контроль проводиться у формі іспиту. При цьому при виведенні остаточної

оцінки враховуються результати поточного контролю.

(ОЦЕ ЗАБРАТИ) Перед вивченням дисципліни, як правило, проводиться вхідний контроль знань з дисциплін, що їй передують і забезпечують. При цьому необхідно встановити рівні та критерії сформованості знань щодо змісту навчальних елементів. Такими рівнями є:

Ознайомчо-орієнтовний (ОО) – особа має орієнтовне уявлення щодо понять, які вивчаються, здатна: принципи об'єктно-орієнтованого програмування, основні поняття ООП (об'єкти, класи, наслідування, перевантаження, поліморфізм, опрацювання виняткових ситуацій, потоки); мета-класи, делегування, шаблони; навчити об'єктно-орієнтованому підходу до розробки і реалізації прикладних програмних систем.

Понятійно-аналітичний (ПА) – особа має чітке уявлення щодо навчального об'єкту, здатна перенести раніше засвоєні знання на типові ситуації.

Продуктивно-синтетичний (ПС) – особа має глибоке розуміння щодо навчального об'єкту, здатна здійснювати синтез, генерувати нові ідеї та уявлення, переносити раніше засвоєні знання на нетипові, нестандартні ситуації. (ОЦЕ ЗАБРАТИ)

Кожний вид роботи з дисципліни оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих позитивно з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід (іспит), вважається невстигаючим.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання лабораторної та практичної робіт – здійснюється на їх початку; засвоєння теоретичного матеріалу з тем перевіряється тестовим контролем; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної та практичної роботи, та індивідуального завдання згідно з робочою програмою дисципліни і робочим навчальним планом.

При оцінюванні знань студентів викладач керується такими критеріями. Оцінку „відмінно” за шкалою ЄКТС – А, отримує студент за глибоке і повне опанування змісту навчального матеріалу з ООП, в якому він легко орієнтується, понятійного апарату, за уміння зв'язувати теорію з практикою, вирішувати практичні завдання, висловлювати і обґрунтовувати свої судження і конструктивні рішення. Відмінна оцінка передбачає грамотний, логічний виклад відповіді (як в усній, так і в письмовій формі), якісне зовнішнє оформлення. Оцінка "відмінно" виставляється студенту, який глибоко засвоїв конструктивні особливості ООП, та вміє їх раціонально застосувати, знає методики та вміє ними користуватися при проектуванні ПЗ. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку „добре” за шкалою ЄКТС – В, отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом ООП, орієнтування в вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента має будуватись на основі самостійного мислення. Оцінку „добре” отримує студент за правильну відповідь з двома-трьома суттєвими помилками.

Оцінки "задовільно" за шкалою ЄКТС – С, заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент слабо знає структуру курсу, допускає помилки у відповіді, засвоїв і набув практичні навички з проектування ПЗ, але допустив неточності. Вагається при відповіді на видозмінене запитання, разом з тим студент володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді. Оцінки "задовільно" заслуговує студент за неповне опанування програмного матеріалу, але отримані знання і набуті практичні навички з розроблення ПЗ відповідають мінімальним критеріям оцінювання.

Оцінка „незадовільно” за шкалою ЄКТС – Е, виставляється, коли студент має розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка "незадовільно" за шкалою ЄКТС – FХ, виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

Як результати навчання, отримані у неформальній освіті, зокрема онлайн-курси «Object-Oriented Programming» (<https://www.edx.org/course/object-oriented-programming>) може бути зараховано виконання двох лабораторних робіт.

На основі результатів поточного контролю і підсумкового контрольного заходу виставляється підсумкова семестрова оцінка. На основі аналізу контролю знань викладач удосконалює курс лекцій, звертаючи особливу увагу на ті розділи, чи теми, з яких було найбільше неточних відповідей, що свідчить про методичні чи інші недоліки при висвітленні вказаних тем або розділів. Аналогічно вносяться корективи в методичні посібники для лабораторних та практичних робіт, детальніше розглядаються принципи питань при виконанні лабораторних та практичних робіт та їх захисті.

### Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота								Самостійна, індивідуальна робота	Форма семестрового контролю
ІІІ семестр									
Лабораторні та практичні роботи №:								Тестовий контроль:	іспит
1	2	3	4	5	6	7	8	Т 1-16	1
ВК: 0,4								0,2	0,4

Примітка: Т – тема дисципліни; ВК – ваговий коефіцієнт;

Для переходу від вітчизняної оцінки до оцінки за шкалою ECTS необхідно знайти середньоарифметичну оцінку за вітчизняною шкалою, помножити її на відповідний ваговий коефіцієнт і, додавши всі складові, отримаємо суму балів, які визначають конкретну оцінку ECTS.

### Оцінювання тестових завдань

Тематичний тест для кожного студента складається з 25 тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 25. Оцінювання здійснюється за чотирибальною шкалою. Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестове завдання	1–13	14–16	17–22	23–25
Оцінка	2	3	4	5

Тестування проводиться з використанням модульного середовища для навчання MOODLE. Правильні відповіді студент реєструє в он-лайн режимі в модульному середовищі MOODLE. Викладач виставляє результати тестування згідно журналу оцінок модульного середовища MOODLE. Підсумкова семестрова оцінка за національною шкалою і шкалою ECTS встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ECTS наведені у наступній таблиці.

### Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ECTS

Оцінка ECTS	Інституційна шкала балів	Інституційна оцінка	Критерії оцінювання	
A	4,75-5,00	5	Зараховано	
B	4,25-4,74	4		<b>Відмінно</b> – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків.
C	3,75-4,24	4		<b>Добре</b> – повне знання навчального матеріалу з кількома незначними помилками.
D	3,25-3,74	3		<b>Добре</b> – в загальному правильна відповідь з двома-трьома суттєвими помилками.
E	3,00-3,24	3		<b>Задовільно</b> – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією.
			<b>Задовільно</b> – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання	

FХ	2,00-2,99	2	Незараховано	<b>Незадовільно</b> – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1,99	2		<b>Незадовільно</b> – необхідна серйозна подальша робота і повторне вивчення дисципліни.

## 5. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ СТУДЕНТІВ ФОРМ НАВЧАННЯ

1. Загальні відомості про програми на С++. Структура проектів в С++.
2. Елементи мови. Службові слова. Імена. Коментарі.
3. Розміщення даних у пам'яті. Тип даних. Адресний вираз.
4. Функції. Область існування імені. Область видимості та простору імен.
5. Типи. Базові стандартні типи. Константи. Змінні.
6. Операції С++. Арифметичні операції. Операції присвоєння. Операції відношення та еквівалентності. Логічні операції.
7. Операції адресації та непрямой адресації.
8. Пріоритет і асоціативність операцій. Перегрузка операцій.
9. Оператори передачі управління. Умовний оператор множинного вибору.
10. Оператори циклів: for, do...while, while.
11. Переривання циклу: оператори break, continue, return, функція Abort.
12. Умовна компіляція. Операції препроцесора.
13. Масиви у С++. Операції з масивами.
14. Правила роботи з масивами.
15. Структури та об'єднання. Структури в стилі С++.
16. Об'єднання. Об'єднання, що не мають імені. Доступ до членів-даних структури, об'єднання.
17. Екземпляри структури, об'єднання. Масиви структур.
18. Вказівники, зсилки та функції в С++.
19. Локальний та динамічний розподіл пам'яті. Функції malloc, calloc, realloc, free.
20. Оператори new и delete. Динамічне розміщення об'єктів та вказівників.
21. Правила роботи з вказівниками. Розіменування вказівників.
22. Оператори new[], оператори delete[].
23. Функції в С++. Передача аргументів та повернення результату.
24. Застосування при передачі параметрів специфікації const. Параметри зі значеннями по умовчанняю.
25. Перегрузка функцій. Приведення типів.
26. Функції-члени класу. Передача у функції змінного числа параметрів.
27. Вбудовані функції inline.
28. Шаблони функцій.
29. Область видимості функцій. Правила, що визначають область видимості.
30. Функція main(). Аргументи функції main. Аргументи командного рядка.
31. Основні поняття ООП. Інкапсуляція, поліморфізм, наслідування і засоби їх реалізації.
32. Об'єкт, властивості, методи, події.
33. Класи. Визначення класу. Ідентифікатори класу. Тіло класу.
34. Приховування інформації. Ієрархії класів та наслідування.
35. Інкапсуляція. Керування доступом до класу. Приватні, загальнодоступні, захищені члени класу.
36. Класи пам'яті для об'єктів класів. Область видимості класу. Порожні класи.
37. Вкладені класи. Правила доступу для вкладених класів.
38. Екземпляри класу. Використання членів даних. Статичні члени-данні.
39. Об'єкти класу як члени даних. Вказівники як члени даних.
40. Використання функцій-членів. Прості, статичні функції члени.
41. Конструктори. Деструктори.

42. Дружні функції. Властивості дружніх функцій.
43. Шаблони класів. Вкладені шаблонні класи.
44. Наслідування. Обмеження наслідування. в C++. Просте наслідування.
45. Специфікатори доступу базових класів. Класи для наслідування. Аргументи передані в базовий клас.
46. Порядок виклику конструкторів. Порядок виклику деструкторів.
47. Перетворення типів у похідних класах. Дозвіл області видимості.
48. Множинне наслідування. Оголошення класу з декількома базовими класами. Виклик конструкторів базових класів.
49. Використання віртуальних базових класів. Використання віртуальних і невіртуальних базових класів разом.
50. Використання перетворення типів. Правила виклику функцій базових класів.
51. Використання дозволу видимості при множинному наслідуванні.
52. Поліморфізм. Раннє та пізнє зв'язування. Віртуальні функції. Перевизначення функції.
53. Абстрактні класи. Обмеження віртуальних функцій. Віртуальні оператори.
54. Поліморфізм при простому наслідуванні. Поліморфізм при множинному наслідуванні.
55. Виклик поліморфних функцій базового класу. Віртуальні функції та ієрархії класифікацій. Виклик віртуальних функцій у конструкторах.
56. Потік C++. Потіки як узагальнені фільтри. Стандартний потоковий ввід-вивід класів користувача.
57. Потіки і файловий ввід-вивід. Використання текстових файлів для введення. Вивід текстових файлів.
58. Обробка подій. Об'єктно-орієнтована програма, яка керована подіями. Методи опрацювання подій.
59. Шаблони функцій і класів. Основні властивості шаблонів класів. Компонентні функції параметризованих класів.
60. Стандартна бібліотека шаблонів. Склад STL. Класи-контейнери. Асоціативні контейнери. Алгоритми.

## 6. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни забезпечений необхідними навчально-методичними розробками в модульному середовищі.

## 7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Олексій Васильєв. Програмування C++ в прикладах і задачах. Ліра-К. 2017. с. 382
2. Галкін О.В., Верес М.М. Мова програмування C++. ДП «Видавничий дім «Персонал» .2017. с.260
3. О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, О. В. Задерейко. C++. Алгоритмізація та програмування. 2-ге вид. перероб. і доповн. Фенікс, 2019. с.477
4. В.В. БУБЛИК. Об'єктно-орієнтоване програмування. Київ. ІТ-книга. 2015. с.640
5. Гнатів, Б. В. - Програмування на С (C++). Парадигма процедурного програмування. Львів. 2017. с.262
6. Ковалюк Т.В. С/C++.Алгоритмізація та програмування. Магнолія 2006. 2021. с.400
7. Козак Л.І. Костюк І.В. Стасевич С.П. Основи програмування. «Новий Світ – 2000» .2020. с.324
8. Томас Г.Кормен. Алгоритми доступно. 2021. с.206
9. В.А. Данілова. Об'єктно-орієнтоване програмування. Практикум. Київ КПІ ім. Ігоря Сікорського 2021. с.121
10. О.Г. Трофименко, Ю.В. Прокоп, І.Г. Швайко, Л.М. Буката, Л.А. Косирева, Ю.Г. Леонов, В.В. Ясинський; за ред. О.Г. Трофименко. Фенікс, 2010. с. 544
11. Галісеєв Г.В. Системне програмування. Університет "Україна".2019. с.113
12. АвторБ'ярне Страуструп.Екскурсія по C++ (2-е видання) (C++ In-Depth Series) 2-е видання. Addison-Wesley Professional. 2018. С.256 (англ)
13. Андріан Келер, Гері Брадскі. Вивчення OpenCV 3: комп'ютерний зір у C++ з бібліотекою OpenCV 1st Edition. O'Reilly Media .2017 . с.1024 (англ)
14. Курт Гюртеон. Оптимізований C++ (Optimized C++). O'Reilly Media. 2016. С.338 (англ)
15. Б'ярне Страуступ .Programming: Principles and Practice Using C++. Addison-Wesley Professional. с.1312 (англ).

## 8. ІНФОРМАЦІЙНІ РЕСУРСИ

### Електронний університет:

1. Модульне середовище для навчання (розміщені усі необхідні матеріали з дисципліни, в тому числі тестові завдання для поточного та семестрового контролю знань).
2. Електронна бібліотека університету.