

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**ІМЕНІ В. О. СУХОМЛІНСЬКОГО**  
Механіко-математичний факультет  
Кафедра інформаційних технологій



**ЗАТВЕРДЖУЮ**

Проректор із науково-педагогічної роботи

О. А. Кузнецова

27 серпня 2020 р.

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**  
**ПРОГРАМУВАННЯ**

Ступінь бакалавра

Галузь знань 12 Інформаційні технології

спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

2020 – 2021 навчальний рік



### **Анотація**

Метою вивчення дисципліни «Програмування», з позиції загальних завдань, є підготовка випускників університету, здатних до вирішення практичних завдань у галузі розробки програмного забезпечення на основі володіння фундаментальними підходами в галузі комп'ютерних наук; задоволення потреб особистості в інтелектуальному і культурному розвитку шляхом отримання вищої освіти в області інформаційних технологій; задоволення потреб суспільства у кваліфікованих кадрах шляхом підготовки фахівців у галузі комп'ютерних наук. Дисципліна, що вивчається, відноситься до циклу професійно зорієнтованих дисциплін; при її складанні визначено особливості підготовки бакалавра, проаналізована нормативно-правова база підготовки (освітньо-професійна програма, вимоги до складання навчальної та робочої програм); досліджено систему викладання (мети, принципи, методи, прийоми та засоби навчання), застосування інформаційно-комунікаційного комплексу; сформована система аудиторних занять (лекція, лабораторне заняття), розроблена технологія організації самостійної роботи студентів.

**Ключові слова:** програмування, мови програмування, технологія навчання, аудиторні заняття, позааудиторна робота, науково-дослідницька робота.

### **Annotation**

From the general point of view, the purpose of studying the discipline "Programming" is to prepare university graduates for solving practical software development problems based on the knowledge of basic approaches in the computer science field; to meet the needs of the individual in intellectual and cultural development by obtaining higher education in the field of information technology; to meet the needs of society in qualified personnel by training specialists in the field of computer science.

The discipline in question belongs to a cycle of professionally oriented disciplines. When it was drawn up, the features of the preparation of a bachelor were determined, the legal and regulatory framework for training (educational and professional program, requirements for the preparation of educational and work programs) was analyzed; the teaching system (goals, principles, methods, techniques and means of teaching) and the use of information and communication complex were studied; a system of classroom lessons (lecture, laboratory lesson) has been formed, a technology for organizing students independent work has been developed.

**Key words:** programming, programming languages, teaching technology, classroom studies, extracurricular work, research work.

Програма вивчення нормативної дисципліни «Програмування» складена відповідно до освітньо-професійної програми підготовки бакалаврів спеціальності 122 Комп'ютерні науки за освітньо-професійною програмою Комп'ютерні науки.

**Предметом вивчення** навчальної дисципліни є об'єктно-орієнтоване програмування мовою C++.

**Міждисциплінарні зв'язки:** алгоритми та структури даних, дискретна математика, теорія графів, обчислення матриць, лінійна алгебра, аналітична геометрія.

## 1. Мета та завдання навчальної дисципліни та очікувані результати

1.1. *Мета курсу* «Програмування» є ознайомлення студентів із основами ООП, поняттями про його базові концепції (інкапсуляція, поліморфізм, наслідування), застосування в подальшому мов програмування, що підтримують методологію ООП (C++, Java, C#). Формування у студентів знань, вмінь і навичок, необхідних для ефективного використання сучасних технологій програмування для розв'язку різних задач та створення прикладного програмного забезпечення загального та професійного спрямування у своїй майбутній професійній, практичній діяльності. У підсумку, вивчення даної дисципліни студент отримує основу для подальшого вивчення усього циклу «комп'ютерних» дисциплін та самостійного оволодіння будь-якими мовами програмування, в основі яких лежить методологія ООП.

1.2. *Основними завданнями* вивчення дисципліни є: набуття знань та навичок у сфері розробки та реалізації алгоритмів мовою програмування C++

### **Програмні результати навчання:**

ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПР2. Використовувати сучасний математичний апарат неперервного та дискретного аналізу, лінійної алгебри, аналітичної геометрії, в професійній діяльності для розв'язання задач теоретичного та прикладного характеру в процесі проектування та реалізації об'єктів інформатизації.

ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

1.3. Згідно з вимогами освітньо-професійної програми студент оволодіває такими компетентностями:

### **I. Загальнопредметні:**

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК9. Здатність працювати в команді.

### **II. Фахові:**

ФК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

ФК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального,

логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

## **2. Інформаційний обсяг навчальної дисципліни.**

**Тема 1.** Огляд об'єктно-орієнтованого програмування. Історія програмування. Модель об'єктно-орієнтованого програмування. Основні принципи ООП. Абстракція даних. Інкапсуляція. Наслідування. Поліморфізм. Методологія ООП.

**Тема 2.** Класи та об'єкти. Синтаксис оголошення класу. Приклад оголошення класу. Визначення функції-члена класу. Створення об'єкта класу. Використання вказівників на об'єкт. Динамічні об'єкти. Використання операції розширення видимості. Обмеження на дані члени-класу. Неповне оголошення класу. Специфікатори доступу. Створення класу в середовищі розробки Code Blocks.

**Тема 3.** Елементи класів. Функції-члени класу.

Конструктори і деструктори. Правила для конструкторів. Правила для деструкторів. Список ініціалізації елементів. Конструктори по замовчуванню. Конструктори копіювання. Вказівник this. Вбудовані функції (inline). Статичні дані-члени класу. Статичні функції-члени класу. Константні дані-члени класу. Константні функції-члени класу. Константні об'єкти.

**Тема 4.** Дружні функції. Дружні класи.

Функції дружні одному класу. Функції дружні декільком класам. Функції-члени дружні іншому класу.

**Тема 5.** Реалізація принципу наслідування у ООП. Просте наслідування.

Робота конструктора і деструктора. Базовий клас координат. Похідний клас точки. Похідний клас вектора. Використання конструктора з параметрами. Перевизначення і виклик членів базового класу.

**Тема 6.** Множинне наслідування.

Робота конструктора і деструктора. Приклад множинного наслідування. Віртуальні базові класи.

**Тема 7.** Механізм перевантаження у ООП. Перевантаження функцій.

Призначення перевантаження. Декорування компілятором імен функцій. Перевантаження функцій. Перевантаження конструкторів. Неоднозначність перевантаження.

**Тема 8.** Перевантаження операторів..

Поняття перевантаження операторів. Правила перевантаження операторів. Перевантаження унарних операторів. Перевантаження оператора присвоювання. Перевантаження бінарних операторів. Перевантаження операторів X=. Перевантаження бінарних операторів, які використовують об'єкти двох класів. Перевантаження операторів інкременту і декременту. Перевантаження оператора індексування. Перевантаження оператора виклику функції. Перетворення типу.

**Тема 9.** Принцип поліморфізму та обробка виключень. Віртуальні функції.

Раннє і пізнє зв'язування. Динамічний поліморфізм. Віртуальні функції. Віртуальні деструктори. Абстрактні класи і чисто віртуальні функції.

**Тема 10.** Потокове введення – виведення. Обробка виключень.

Зумовлені потоки. Переваги потоків бібліотеки C++. Операції поміщення і витягання із потоку. Форматування потоку. Файлове введення – виведення з використанням потоків. Обробка виключних ситуацій. Генерування виключень.

### 3. Рекомендована література

#### Базова

1. Роберт Лафоре. Объектно-ориентированное программирование в С++ - Питер, 2013. 928 с.
2. Бондарев В.М. Программирование на С++ - Харьков: «Компания СМІТ», 2005. 284 с.
3. В.В. Мухортов, В.Ю. Рылов. Объектно-Ориентированное Программирование, Анализ и Дизайн. Новосибирск, 2006. 103 с.
4. Тимоти Бадд. Объектно-ориентированное программирование в действии. 2006.
5. Астахова И.Ф., Власов С.В. Язык С++. Учебное пособие. Мн: Новое название, 2003. 203 с.
6. Александреску А. Современное проектирование на С++ М: Издательский дом «Вильямс», 2002. 336 с.

#### Допоміжна

1. Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влссидес: Приемы ООП. Паттерны проектирования. Питер, 2001. 368 с.
2. Астахова И.Ф., Власов С.В. - Язык С++. Учебное пособие. Мн: Новое название, 2003. 203 с.
3. Хэнкон Л., Кригер М. Введение в программирование на языке Си. М.: Сов. радио, 1986.
4. Белецкий Я. Энциклопедия языка Си. М.: Мир, 1992.
5. С++. Язык программирования. М.: «И.В.К.-СОФТ», 1991.
6. Березин В.Н., Березин И.В. Начальный курс Си и Си++. М.: Диалог-Мифи, 1996.
7. Иан Грэхем Объектно-ориентированные методы. Принципы и практика = Object-Oriented Methods: Principles & Practice. 3-е изд. М.: «Вильямс», 2004. 880 с.
8. Антони Синтес Освой самостоятельно объектно-ориентированное программирование за 21 день = Sams Teach Yourself Object-Oriented Programming in 21 Days. М.: «Вильямс», 2002. 672 с.
9. Ашарина, И.В. Основы программирования на языках С и С++ / И.В. Ашарина. М.: ГЛТ, 2012. 208 с.
10. Вирт Н. Алгоритмы и структуры данных. - М.: Мир, 1989. 360 с.
11. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. М.: Вильямс, 2000
12. М. Эллис, Б. Страуструп. Справочное руководство по языку С++ с комментариями: Пер. с англ. Москва: Мир, 1992. 445с.
13. Стенли Б. Липпман. С++ для начинающих: Пер. с англ. Москва: Унитех; Рязань: Гэлион, 1992. 345с.
14. Бруно Бабэ. Просто и ясно о Borland С++: Пер. с англ. Москва: БИНОМ, 1994. 400с.
15. В. Липский Комбинаторика для программистов. 1988. 200 с.
16. С.Окулов Программирование в алгоритмах. 2006. 220с.
17. Серджвик Р. Фундаментальные алгоритмы на С. Часть 1-3 СПб: Диасофт, 2003.
18. Кениг, Э. Эффективное программирование на С++. Практическое программирование на примерах. Т. 2 / Э. Кениг, Б.Э. Му. - М.: Вильямс, 2016. 368 с.
19. МакГрат, М. Программирование на С для начинающих / М. МакГрат. М.: Эксмо, 2015. 192 с.
20. Мартынов, Н.Н. Программирование для Windows на С / Н.Н. Мартынов. М.: БИНОМ, 2013. 528 с.
21. Перри, Г. Программирование на С для начинающих / Г. Перри, Д. Миллер. М.: Эксмо, 2015. 368 с.
22. 9. Фленов, М.Е. Программирование на С++ глазами хакера. / М.Е. Фленов. СПб.: ВHV, 2012. 352 с.
23. Хенкеманс, Д. Программирование на С++ / Д. Хенкеманс, М. Ли. СПб.: Символ-плюс, 2015. 416 с.
24. Бочков С.О. Субботин Д.М. Язык программирования С для персонального компьютера. М.: Радио и связь, 1990.
25. В.В. Подбельский. Язык С++: Учебное пособие. Москва: Финансы и статистика, 1995. 560с.

26. Г. Шилдт. Самоучитель С++: Пер. с англ. Санкт-Петербург: ВHV-Санкт-Петербург, 1998. 620с.
27. У. Сэвитч. С++ в примерах: Пер. с англ. Москва: ЭКОМ, 1997. 736с.
28. К. Джамса. Учимся программировать на языке С++: Пер. с англ. Москва: Мир, 1997. 320с.
29. Керниган Б., Ритчи Д. Язык программирования С. М.: Финансы и статистика, 1985.
30. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования С. Задачи по языку С. М.: Финансы и статистика, 1985.
31. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. М.: Вильямс, 2006.
32. Абельсон Х., Сассман Д. Структура и интерпретация компьютерных программ. М.: Добросвет, 2006
33. Уильям Топп, Уильям Форд. Структуры данных в С++: Пер. с англ. М.: ЗАО «Издательство БИНОМ», 2000. 816 с.
34. Караванова Т.П. Информатика. Методи побудови алгоритмів та їх аналіз. К.: Генеза. 2007. 216 с.
35. Дорогов, В.Г. Основы программирования на языке С: Учебное пособие / В.Г. Дорогов, Е.Г. Дорогова; Под общ. ред. проф. Л.Г. Гагарина. М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. 224 с.

### Інформаційні ресурси

1. Объектно-ориентированное программирование. URL: <http://ru.wikipedia.org/wiki/> (дата звернення 23.09.2019)
2. Основы ООП. URL: <http://www.youtube.com/watch?v=rbCkcBDobCY> (дата звернення 03.12.2018)
3. Три кита объектно-ориентированного программирования. URL: <http://www.codenet.ru/progr/vbasic/oop.php> (дата звернення 15.05.2020)
4. Object-Oriented Programming (C# and Visual Basic). URL: <http://msdn.microsoft.com/ru-ru/library/dd460654.aspx> (дата звернення 25.11.2018)
5. Принципы объектно-ориентированного программирования. URL: <http://www.weblibrary.biz/c-sharp/principy> (дата звернення 14.10.2018)
6. Объектно-ориентированные языки программирования. URL: <http://proger.elitno.net/oop/121-preimushchestva-i-nedostatki-oop-obektno-orientirovannye-yazyki-programmirovaniya> (дата звернення 12.10.2019)

#### 4. Форма підсумкового контролю успішності навчання: залік

##### 5. Засоби діагностики успішності навчання:

Лекційні та лабораторні заняття з використанням традиційних методів, інтерактивних методик та комп'ютерної техніки, індивідуальна робота, самостійна робота.

Поточний та підсумковий контроль здійснюється у вигляді комп'ютерних тестів на освітньому просторі університету. Для оцінювання використовується національна чотирьохбальна шкала: відмінно, добре, задовільно, незадовільно; європейська шкала: А, В, С, D, E, FX, F.

(приклад для заліку) 100% балів студенти накопичують на заняттях та під час поточного і підсумкового контролю, що регламентується робочою програмою викладача. (приклад для іспиту) 60% балів студенти накопичують на заняттях та під час поточного контролю, що регламентується робочою програмою викладача, 40% балів студенти набирають на заліку.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**ІМЕНІ В. О. СУХОМЛИНСЬКОГО**  
Механіко-математичний факультет  
Кафедра інформаційних технологій

**ЗАТВЕРДЖУЮ**

Проректор із науково-педагогічної роботи

\_\_\_\_\_ О. А. Кузнецова

27 серпня 2020 р.

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**  
**ПРОГРАМУВАННЯ**

Ступінь бакалавра

Галузь знань 12 Інформаційні технології

спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

2020 – 2021 навчальний рік



Розробник: Ходякова Галина Вікторівна, доцент кафедри інформаційних технологій, кандидат педагогічних наук \_\_\_\_\_ (Ходякова Г.В.)

Робоча програма затверджена на засіданні кафедри інформаційних технологій

Протокол № 1 від «26» серпня 2020 р.

Завідувач кафедри \_\_\_\_\_ (Зосімов В.В.)

«26» серпня 2020 р.

Програму погоджено з гарантом ОП Комп'ютерні науки

доцент кафедри, к.техн.н. \_\_\_\_\_ (Булгакова О.С.)

### **Анотація**

Метою вивчення дисципліни «Програмування», з позиції загальних завдань, є підготовка випускників університету, здатних до вирішення практичних завдань у галузі розробки програмного забезпечення на основі володіння фундаментальними підходами в галузі комп'ютерних наук; задоволення потреб особистості в інтелектуальному і культурному розвитку шляхом отримання вищої освіти в області інформаційних технологій; задоволення потреб суспільства у кваліфікованих кадрах шляхом підготовки фахівців у галузі комп'ютерних наук. Дисципліна, що вивчається, відноситься до циклу професійно зорієнтованих дисциплін; при її складанні визначено особливості підготовки бакалавра, проаналізована нормативно-правова база підготовки (освітньо-професійна програма, вимоги до складання навчальної та робочої програм); досліджено систему викладання (мети, принципи, методи, прийоми та засоби навчання), застосування інформаційно-комунікаційного комплексу; сформована система аудиторних занять (лекція, лабораторне заняття), розроблена технологія організації самостійної роботи студентів.

**Ключові слова:** програмування, мови програмування, технологія навчання, аудиторні заняття, позааудиторна робота, науково-дослідницька робота.

### **Annotation**

From the general point of view, the purpose of studying the discipline "Programming" is to prepare university graduates for solving practical software development problems based on the knowledge of basic approaches in the computer science field; to meet the needs of the individual in intellectual and cultural development by obtaining higher education in the field of information technology; to meet the needs of society in qualified personnel by training specialists in the field of computer science.

The discipline in question belongs to a cycle of professionally oriented disciplines. When it was drawn up, the features of the preparation of a bachelor were determined, the legal and regulatory framework for training (educational and professional program, requirements for the preparation of educational and work programs) was analyzed; the teaching system (goals, principles, methods, techniques and means of teaching) and the use of information and communication complex were studied; a system of classroom lessons (lecture, laboratory lesson) has been formed, a technology for organizing students independent work has been developed.

**Key words:** programming, programming languages, teaching technology, classroom studies, extracurricular work, research work.

## 1. Опис навчальної дисципліни

Денна форма навчання

Найменування показників	Галузь знань, освітній ступінь	Характеристика навчальної дисципліни	
		<i>денна форма навчання</i>	
Кількість кредитів – 5	Галузь знань 12 Інформаційні технології	Нормативна	
Індивідуальне науково-дослідне завдання – проєкт, наукова стаття  Загальна кількість годин – 150	Спеціальність 122 Комп'ютерні науки	<b><i>Семестр</i></b>	
		5-й	
		<b><i>Лекції</i></b>	
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 6	Ступінь бакалавра	<b><i>16 год.</i></b>	
		<b><i>Практичні, семінарські</i></b>	
		<b><i>Лабораторні</i></b>	
		<b><i>34 год.</i></b>	
<a href="http://moodle.mdu.edu.ua/my/">http://moodle.mdu.edu.ua/my/</a>		<b><i>Самостійна робота</i></b>	
		<b><i>100 год.</i></b>	
		<b><i>Вид контролю: залік</i></b>	

Мова навчання – українська.

**Примітка.** Співвідношення кількості годин аудиторних занять до самостійної та індивідуальної роботи становить: для денної форми навчання – 50 год. – аудиторні заняття, 100 год.– самостійна робота (33% / 67%).

## **2. Мета, завдання навчальної дисципліни та очікувані результати**

*Мета курсу:* формування у студентів знань, вмінь і навичок, необхідних для ефективного використання сучасних технологій програмування для розв'язку різних задач.

*Завдання курсу:* набуття знань та навичок у сфері розробки та реалізації алгоритмів мовою програмування C++

### **Передумови для вивчення дисципліни:**

Навчальна дисципліна складається з 5-ти кредитів.

### **Програмні результати навчання:**

ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПР2. Використовувати сучасний математичний апарат неперервного та дискретного аналізу, лінійної алгебри, аналітичної геометрії, в професійній діяльності для розв'язання задач теоретичного та прикладного характеру в процесі проектування та реалізації об'єктів інформатизації.

ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

Згідно з вимогами освітньо-професійної програми студент оволодіває такими компетентностями:

### **I. Загальнопредметні:**

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК9. Здатність працювати в команді.

### **II. Фахові:**

ФК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

ФК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

### **Інформаційний обсяг навчальної дисципліни.**

**Тема 1.** Огляд об'єктно-орієнтованого програмування. Історія програмування. Модель об'єктно-орієнтованого програмування. Основні принципи ООП. Абстракція даних. Інкапсуляція. Наслідування. Поліморфізм. Методологія ООП.

**Тема 2.** Класи та об'єкти. Синтаксис оголошення класу. Приклад оголошення класу. Визначення функції-члена класу. Створення об'єкта класу. Використання вказівників на об'єкт. Динамічні об'єкти. Використання операції розширення видимості. Обмеження на дані члени-класу. Неповне оголошення класу. Специфікатори доступу. Створення класу в середовищі розробки Code Blocks.

**Тема 3.** Елементи класів. Функції-члени класу.

Конструктори і деструктори. Правила для конструкторів. Правила для деструкторів. Список ініціалізації елементів. Конструктори по замовчуванню. Конструктори копіювання. Вказівник this. Вбудовані функції (inline). Статичні дані-члени класу. Статичні функції-члени класу. Константні дані-члени класу. Константні функції-члени класу. Константні об'єкти.

**Тема 4.** Дружні функції. Дружні класи.

Функції дружні одному класу. Функції дружні декільком класам. Функції-члени дружні іншому класу.

**Тема 5.** Реалізація принципу наслідування у ООП. Просте наслідування.

Робота конструктора і деструктора. Базовий клас координат. Похідний клас точки. Похідний клас вектора. Використання конструктора з параметрами. Перевизначення і виклик членів базового класу.

**Тема 6.** Множинне наслідування.

Робота конструктора і деструктора. Приклад множинного наслідування. Віртуальні базові класи.

**Тема 7.** Механізм перевантаження у ООП. Перевантаження функцій.

Призначення перевантаження. Декорування компілятором імен функцій. Перевантаження функцій. Перевантаження конструкторів. Неоднозначність перевантаження.

**Тема 8.** Перевантаження операторів.

Поняття перевантаження операторів. Правила перевантаження операторів. Перевантаження унарних операторів. Перевантаження оператора присвоювання. Перевантаження бінарних операторів. Перевантаження операторів  $X=$ . Перевантаження бінарних операторів, які використовують об'єкти двох класів. Перевантаження операторів інкременту і декременту. Перевантаження оператора індексування. Перевантаження оператора виклику функції. Перетворення типу.

**Тема 9.** Принцип поліморфізму та обробка виключень. Віртуальні функції.

Раннє і пізнє зв'язування. Динамічний поліморфізм. Віртуальні функції. Віртуальні деструктори. Абстрактні класи і чисто віртуальні функції.

**Тема 10.** Потокове введення – виведення. Обробка виключень.

Зумовлені потоки. Переваги потоків бібліотеки C++. Операції поміщення і витягання із потоку. Форматування потоку. Файлове введення – виведення з використанням потоків. Обробка виключних ситуацій. Генерування виключень.

## Програма навчальної дисципліни

### Кредит 1. Основні принципи ООП. Класи та об'єкти.

**Тема 1.** Огляд об'єктно-орієнтованого програмування. Історія програмування. Модель об'єктно-орієнтованого програмування. Основні принципи ООП. Абстракція даних. Інкапсуляція. Наслідування. Поліморфізм. Методологія ООП.

**Тема 2.** Класи та об'єкти. Синтаксис оголошення класу. Приклад оголошення класу. Визначення функції-члена класу. Створення об'єкта класу. Використання вказівників на об'єкт. Динамічні об'єкти. Використання операції розширення видимості. Обмеження на дані члени-класу. Неповне оголошення класу. Специфікатори доступу. Створення класу в середовищі розробки Code Blocks.

### Кредит 2. Елементи класів.

#### Тема 3. Функції-члени класу.

Конструктори і деструктори. Правила для конструкторів. Правила для деструкторів. Список ініціалізації елементів. Конструктори по замовчуванню. Конструктори копіювання. Вказівник this. Вбудовані функції (inline). Статичні дані-члени класу. Статичні функції-члени класу. Константні дані-члени класу. Константні функції-члени класу. Константні об'єкти.

#### Тема 4. Дружні функції. Дружні класи.

Функції дружні одному класу. Функції дружні декільком класам. Функції-члени дружні іншому класу.

### Кредит 3. Реалізація принципу наслідування у ООП.

#### Тема 5. Просте наслідування.

Робота конструктора і деструктора. Базовий клас координат. Похідний клас точки. Похідний клас вектора. Використання конструктора з параметрами. Перевизначення і виклик членів базового класу.

#### Тема 6. Множинне наслідування.

Робота конструктора і деструктора. Приклад множинного наслідування. Віртуальні базові класи.

### Кредит 4. Механізм перевантаження у ООП

#### Тема 7. Перевантаження функцій.

Призначення перевантаження. Декорування компілятором імен функцій. Перевантаження функцій. Перевантаження конструкторів. Неоднозначність перевантаження.

#### Тема 8. Перевантаження операторів..

Поняття перевантаження операторів. Правила перевантаження операторів. Перевантаження унарних операторів. Перевантаження оператора присвоювання. Перевантаження бінарних операторів. Перевантаження операторів X=. Перевантаження бінарних операторів, які використовують об'єкти двох класів. Перевантаження операторів інкременту і декременту. Перевантаження оператора індексування. Перевантаження оператора виклику функції. Перетворення типу.

### Кредит 5. Принцип поліморфізму та обробка виключень.

#### Тема 9. Віртуальні функції.

Раннє і пізнє зв'язування. Динамічний поліморфізм. Віртуальні функції. Віртуальні деструктори. Абстрактні класи і чисто віртуальні функції.

#### Тема 10. Потокове введення – виведення. Обробка виключень.

Зумовлені потоки. Переваги потоків бібліотеки C++. Операції поміщення і витягання із потоку. Форматування потоку. Файлове введення – виведення з використанням потоків. Обробка виключних ситуацій. Генерування виключень.

### 3. Структура навчальної дисципліни Денна форма навчання

Назви змістових модулів і тем	Кількість годин					
	Денна форма					
	Усього	у тому числі				
		л	п	лаб	інд	ср
<b>Кредит 1. Основні принципи ООП. Класи та об'єкти.</b>						
Тема 1. Огляд об'єктно-орієнтованого програмування.	14	2		2		10
Тема 2. Класи.	16	2		4		10
Усього	30	4		6		20
<b>Кредит 2. Елементи класів.</b>						
Тема 3. Функції-члени класу.	14	2		2		10
Тема 4. Дружні функції. Дружні класи.	16	2		4		10
Усього	30	4		6		20
<b>Кредит 3. Реалізація принципу наслідування у ООП.</b>						
Тема 5. Просте наслідування.	16	2		4		10
Тема 6. Множинне наслідування.	14			4		10
Усього	30	2		8		20
<b>Кредит 4. Механізм перевантаження у ООП</b>						
Тема 7. Перевантаження функцій.	14	2		2		10
Тема 8. Перевантаження операторів	16	2		4		10
Усього	30	4		6		20
<b>Кредит 5. Принцип поліморфізму та обробка виключень.</b>						
Тема 9. Віртуальні функції	16	2		4		10
Тема 10. Потокowe введення – виведення. Обробка виключень.	14			4		10
Усього	30	2		8		20
<b>Усього годин:</b>	<b>150</b>	<b>16</b>		<b>34</b>		<b>100</b>

#### 4. Теми лекційних занять

№ з/п	Назва теми	Кількість годин
	<i>Кредит 1. Основні принципи ООП. Класи та об'єкти.</i>	
1	Тема 1. Огляд об'єктно-орієнтованого програмування.	2
2	Тема 2. Класи.	2
	<i>Кредит 2. Елементи класів.</i>	
3	Тема 3. Функції-члени класу.	2
4	Тема 4. Дружні функції. Дружні класи.	2
<i>Кредит 3. Реалізація принципу наслідування у ООП.</i>		
5	Тема 5. Просте наслідування. Множинне наслідування.	2
<i>Кредит 4. Механізм перевантаження у ООП</i>		
6	Тема 6. Перевантаження функцій.	2
7	Тема 7. Перевантаження операторів	2
<i>Кредит 5. Принцип поліморфізму та обробка виключень</i>		
8	Тема 8. Віртуальні функції. Поточкове введення – виведення. Обробка виключень.	2
	Разом:	16

#### 5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
	<i>Кредит 1. Основні принципи ООП. Класи та об'єкти.</i>	
1	Тема 1. Створення проєктів зі структурами.	2
2	Тема 2. Створення класу.	4
	<i>Кредит 2. Елементи класів.</i>	
3	Тема 3. Класи та дані-члени класів	2
4	Тема 4. Функції-члени класів	2



5	Тема 5. Види конструкторів	2
<i>Кредит 3. Реалізація принципу наслідування у ООП.</i>		
6	Тема 6. Розробка класів та їх структурних елементів	4
7	Тема 7. Масиви у класах. Масиви об'єктів	4
<i>Кредит 4. Механізм перевантаження у ООП</i>		
8	Тема 8. Дружні функції. Розробка різних видів-методів класу	2
9	Тема 9. Просте наслідування. Множинне наслідування.	4
<i>Кредит 5. Принцип поліморфізму та обробка виключень</i>		
10	Тема 10. Перевантаження унарних та бінарних операторів.	2
11	Тема 10. Оператори виведення і вставки в потік.	2
12	Тема 12. Файлове введення і виведення.	4
	Разом:	34

## 6. Самостійна робота

№ з/п	Назва теми	Кількість годин
<i>Кредит 1. Основні принципи ООП. Класи та об'єкти.</i>		
1.	Тема 1. Опрацювання структур	7
2.	Тема 2. Види конструкторів	7
	Оформлення звітів. Підготовка до захисту робіт	6
<i>Кредит 2. Елементи класів.</i>		
3.	Тема 3. Перевантаження операцій.	8
4.	Тема 4. Абстрактні класи.	7
	Оформлення звітів. Підготовка до захисту робіт	5
<i>Кредит 3. Реалізація принципу наслідування у ООП.</i>		
5.	Тема 5. Шаблони класів.	6
6.	Тема 6. Опрацювання виключних ситуацій.	6

	Оформлення звітів. Підготовка до захисту робіт	6
<i>Кредит 4. Механізм перевантаження у ООП</i>		
7.	Тема 7. Шаблони класів динамічних масивів.	6
8.	Тема 8. Потоківі класи.	6
	Оформлення звітів. Підготовка до захисту робіт	6
<i>Кредит 5. Принцип поліморфізму та обробка виключень.</i>		
9.	Тема 9. Спеціальні класи, функції і вказівники.	6
10.	Тема 10. Препроцесор і компілятор.	6
	Оформлення звітів. Підготовка до захисту робіт	6
Всього :		100

### 7. Індивідуальне навчально-дослідне завдання

1. проєкт,
2. наукова стаття

### 8. Форми роботи та критерії оцінювання

Рейтинговий контроль знань студентів здійснюється за 100-бальною шкалою:

#### Шкала оцінювання: національна та ECTS

ОЦІНКА ECTS	СУМА БАЛІВ	ОЦІНКА ЗА НАЦІОНАЛЬНОЮ ШКАЛОЮ	
		екзамен	залік
A	90-100	5 (відмінно)	5/відм./зараховано
B	80-89	4 (добре)	4/добре/ зараховано
C	65-79		
D	55-64	3 (задовільно)	3/задов./ зараховано
E	50-54		
FX	35-49	2 (незадовільно)	Не зараховано

**Форми поточного та підсумкового контролю.** Комплексна діагностика знань, умінь і навичок студентів із дисципліни здійснюється на основі результатів проведення поточного й підсумкового контролю знань (КР). Поточне оцінювання (індивідуальне, групове і фронтальне опитування, самостійна робота, самоконтроль). Завданням поточного контролю є систематична перевірка розуміння та засвоєння програмового матеріалу, виконання практичних, лабораторних робіт, уміння самостійно опрацьовувати тексти, складання конспекту рекомендованої літератури, написання і захист реферату, здатності публічно чи письмово представляти певний матеріал.

Завданням підсумкового контролю (КР, залік) є перевірка глибини засвоєння студентом програмового матеріалу модуля.

*Критерії оцінювання відповідей на практичних заняттях:*

Студенту виставляється відмінно, якщо студент здатний самостійно здійснювати основні види навчальної діяльності. Знання студента є глибокими, міцними, узагальненими; студент вміє застосовувати знання творчо, його навчальна діяльність позначена вмінням

самостійно оцінювати різноманітні життєві ситуації, явища, факти, виявляти і відстоювати особисту позицію.

Студенту виставляється дуже добре, якщо студент знає істотні ознаки понять, явищ, закономірностей, зв'язків між ними, а також самостійно застосовує знання в нестандартних ситуаціях, володіє розумовими операціями, вміє робити висновки, виправляти допущені помилки. Відповідь повна, правильна, логічна, обґрунтована.

Студенту виставляється добре, якщо студент знає ознаки понять, явищ, закономірностей, зв'язків між ними на середньому рівні, а також самостійно застосовує знання в стандартних ситуаціях, володіє розумовими операціями, вміє робити висновки, виправляти допущені помилки. Відповідь повна, правильна, логічна, обґрунтована.

Студенту виставляється достатньо, якщо відповідь студента при відтворенні навчального матеріалу елементарна, зумовлюється початковими уявленнями про предмет вивчення. Студент відтворює основний навчальний матеріал, здатний виконувати завдання за зразком, володіє елементарними вміннями навчальної діяльності.

Студенту виставляється мінімальний задовільно, якщо відповідь студента при відтворенні навчального матеріалу елементарна, зумовлюється початковими уявленнями про предмет вивчення. Студент відтворює основний навчальний матеріал.

Кількість балів у кінці **семестру** повинна складати від 250 до 500 балів (за 5 кредитів), тобто сума балів за виконання усіх завдань.

Відповідний **розподіл балів, які отримують студенти за 5 крд**

Поточне оцінювання та самостійна робота										КР	Накоп. бали/ Сума
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	100	500/100
40	40	40	40	40	40	40	40	40	40		

## 9. Засоби діагностики

Засобами діагностики та методами демонстрування результатів навчання є: завдання до лабораторних занять, завдання для самостійної та індивідуальної роботи, презентації результатів досліджень, тестові завдання, контрольні роботи.

## 10. Методи навчання

Курс складається з лекційних, лабораторних занять та самостійної роботи студентів, домашніх завдань і завершується підсумковим рейтинг-контролем і виконанням конкретних контрольних завдань по даній дисципліні.

Лекційні заняття призначені для теоретичного осмислення і узагальнення базових розділів курсу.

Лабораторні заняття є аудиторними, проводяться по наперед відомих темах у вигляді практичної роботи студентів за комп'ютерами. Лабораторне заняття – форма навчального заняття, спрямована на закріплення та вдосконалення студентом теоретичних знань, отриманих як на лекційних заняттях, так і в процесі самостійного вивчення матеріалу. Під час лабораторного заняття студенти під керівництвом викладача особисто набувають практичних навичок у роботі з обчислювальною технікою, оволодівають методикою створення програмних продуктів у програмному середовищі. У ході заняття студенти самостійно виконують передбачені завданням дії, заносючи результати в звіт. На це відводиться до 85 – 90% часу заняття. Викладач здійснює контроль за роботою і надає допомогу при виникненні труднощів, звертає увагу на складні, ключові моменти. Причому основна увага приділяється не вказівці на конкретну помилку, а методиці пошуку причин виникнення цих помилок. Складання звіту – це відповідальний етап лабораторного заняття. При його складанні студенти розвивають навички

аналізу, узагальнення і творчого осмислення результатів роботи, а також навички розробки документації. Кожна лабораторна робота передбачає наявність індивідуального завдання і наступного захисту.

Самостійна робота є позааудиторною і призначена для самостійного ознайомлення студента з певними розділами курсу за рекомендованими педагогом матеріалами і підготовки до виконання індивідуальних завдань по курсу.

Поточний рейтинг-контроль проводиться викладачем в процесі проведення всіх видів занять. Проміжний рейтинг-контроль призначений для практичної комплексної оцінки освоєння розділів курсу і здійснюється шляхом підготовки студентами відповідей на поставлені питання.

Кінцевий контроль знань з дисципліни проводиться під час складання іспиту.

## 11. Рекомендована література

### Базова

1. Роберт Лафоре. Объектно-ориентированное программирование в С++ - Питер, 2013. 928 с.
2. Бондарев В.М. Программирование на С++ - Харьков: «Компания СМІТ», 2005. 284 с.
3. В.В. Мухортов, В.Ю. Рылов. Объектно-Ориентированное Программирование, Анализ и Дизайн. Новосибирск, 2006. 103 с.
4. Тимоти Бадд. Объектно-ориентированное программирование в действии. 2006.
5. Астахова И.Ф., Власов С.В. Язык С++. Учебное пособие. Мн: Новое название, 2003. 203 с.
6. Александреску А. Современное проектирование на С++ М: Издательский дом «Вильямс», 2002. 336 с.

### Допоміжна

1. Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес: Приемы ООП. Паттерны проектирования. Питер, 2001. 368 с.
2. Астахова И.Ф., Власов С.В. - Язык С++. Учебное пособие. Мн: Новое название, 2003. 203 с.
3. Хэнкон Л., Кригер М. Введение в программирование на языке Си. М.: Сов. радио, 1986.
4. Белецкий Я. Энциклопедия языка Си. М.: Мир, 1992.
5. С++. Язык программирования. М.: «И.В.К.-СОФТ», 1991.
6. Березин В.Н., Березин И.В. Начальный курс Си и Си++. М.: Диалог-Мифи, 1996.
7. Иан Грэхем Объектно-ориентированные методы. Принципы и практика = Object-Oriented Methods: Principles & Practice. 3-е изд. М.: «Вильямс», 2004. 880 с.
8. Антони Синтес Освой самостоятельно объектно-ориентированное программирование за 21 день = Sams Teach Yourself Object-Oriented Programming in 21 Days. М.: «Вильямс», 2002. 672 с.
9. Ашарина, И.В. Основы программирования на языках С и С++ / И.В. Ашарина. М.: ГЛТ, 2012. 208 с.
10. Вирт Н. Алгоритмы и структуры данных. - М.: Мир, 1989. 360 с.
11. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. М.: Вильямс, 2000
12. М. Эллис, Б. Страуструп. Справочное руководство по языку С++ с комментариями: Пер. с англ. Москва: Мир, 1992. 445с.
13. Стенли Б. Липпман. С++ для начинающих: Пер. с англ. Москва: Унитех; Рязань: Гэлион, 1992. 345с.
14. Бруно Бабэ. Просто и ясно о Borland С++: Пер. с англ. Москва: БИНОМ, 1994. 400с.
15. В. Липский Комбинаторика для программистов. 1988. 200 с.
16. С.Окулов Программирование в алгоритмах. 2006. 220с.
17. Серджвик Р. Фундаментальные алгоритмы на С. Часть 1-3 СПб: Диасофт, 2003.

18. Кениг, Э. Эффективное программирование на С++. Практическое программирование на примерах. Т. 2 / Э. Кениг, Б.Э. Му. - М.: Вильямс, 2016. 368 с.
19. МакГрат, М. Программирование на С для начинающих / М. МакГрат. М.: Эксмо, 2015. 192 с.
20. Мартынов, Н.Н. Программирование для Windows на С / Н.Н. Мартынов. М.: БИНОМ, 2013. 528 с.
21. Перри, Г. Программирование на С для начинающих / Г. Перри, Д. Миллер. М.: Эксмо, 2015. 368 с.
22. 9. Фленов, М.Е. Программирование на С++ глазами хакера. / М.Е. Фленов. СПб.: BHV, 2012. 352 с.
23. Хенкеманс, Д. Программирование на С++ / Д. Хенкеманс, М. Ли. СПб.: Символ-плюс, 2015. 416 с.
24. Бочков С.О. Субботин Д.М. Язык программирования С для персонального компьютера. М.: Радио и связь, 1990.
25. В.В. Подбельский. Язык С++: Учебное пособие. Москва: Финансы и статистика, 1995. 560с.
26. Г. Шилдт. Самоучитель С++: Пер. с англ. Санкт-Петербург: BHV-Санкт-Петербург, 1998. 620с.
27. У. Сэвитч. С++ в примерах: Пер. с англ. Москва: ЭКОМ, 1997. 736с.
28. К. Джамса. Учимся программировать на языке С++: Пер. с англ. Москва: Мир, 1997. 320с.
29. Керниган Б., Ритчи Д. Язык программирования С. М.: Финансы и статистика, 1985.
30. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования С. Задачи по языку С. М.: Финансы и статистика, 1985.
31. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. М.: Вильямс, 2006.
32. Абельсон Х., Сассман Д. Структура и интерпретация компьютерных программ. М.: Добросвет, 2006
33. Уильям Топп, Уильям Форд. Структуры данных в С++: Пер. с англ. Уильям Топп, Уильям Форд. Структуры данных в С++: Пер. с англ. М.: ЗАО «Издательство БИНОМ», 2000. 816 с.
34. Караванова Т.П. Информатика. Методи побудови алгоритмів та їх аналіз. К.: Генеза. 2007. 216 с.
35. Дорогов, В.Г. Основы программирования на языке С: Учебное пособие / В.Г. Дорогов, Е.Г. Дорогова; Под общ. ред. проф. Л.Г. Гагарина. М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. 224 с.

## 12. Інформаційні ресурси

1. Объектно-ориентированное программирование. URL: <http://ru.wikipedia.org/wiki/> (дата звернення 23.09.2019)
2. Основы ООП. URL: <http://www.youtube.com/watch?v=rbCkcBDobCY> (дата звернення 03.12.2018)
3. Три кита объектно-ориентированного программирования. URL: <http://www.codenet.ru/progr/vbasic/oop.php> (дата звернення 15.05.2020)
4. Object-Oriented Programming (C# and Visual Basic). URL: <http://msdn.microsoft.com/ru-ru/library/dd460654.aspx> (дата звернення 25.11.2018)
5. Принципы объектно-ориентированного программирования. URL: <http://www.weblibrary.biz/c-sharp/princypu> (дата звернення 14.10.2018)
6. Объектно-ориентированные языки программирования. URL: <http://proger.elitno.net/oop/121-preimushchestva-i-nedostatki-oop-obektno-orientirovannye-yazyki-programmirovaniya> (дата звернення 12.10.2019)