

# ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ ТА ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ

**I, II семестри 2024-2025 навч. року, силабус курсу**

освітня програма **Технології штучного інтелекту**

Спеціальність 126 – Інформаційні системи та технології,  
перший (бакалавр) рівень освіти.

**Метою курсу** є формування базових знань та вмінь студента з основ об'єктно-орієнтованого програмування, засобів програмування за допомогою алгоритмічної мови C++, систем проектування та реалізації програмних систем на основі сучасних технологій проектування.

## **Семестр 1.**

**Модуль 1.** Структурне програмування за допомогою C++ (15 годин лекцій, 15 годин лабораторних робіт).

Тема 1. Введення в об'єктно-орієнтоване програмування (ООП). Парадигми програмування. Особливості мови C++. Особливості забудови консольного додатку у середовищі Microsoft Visual Community 2022.

Тема 2. Основи синтаксису і семантики мови C++. Алфавіт і лексеми мови C++ Базові типи даних. Склад виразу на C++.

Тема 3. Структурування дій у програмі на C++. Організація програми на C++. Розподілення пам'яті.

Тема 4. Структурування областей дії змінних в програмі на C++. Область дії змінної в C++. Принципи організації введення-виведення даних в C та C++.

Тема 5. Організація роботи з покажчиками в C++. Поняття покажчика в C++. Робота з динамічною областю пам'яті.

Тема 6. Організація структурованих типів мови C++. Масиви. Рядки. Перерахування. Структури. Об'єднання. Поля біт.

**Модуль 2.** Поняття інкапсуляції в C++ (15 годин лекцій, 15 годин лабораторних робіт).

Тема 7. Інкапсуляція. Частина 1. Основні концепції об'єктно-орієнтованого програмування (ООП). Агрегація. Поняття інкапсуляції, класу та об'єкту. Принципи відділення інтерфейсу від реалізації. Особливість оголошення і використання функцій-членів (методів) класу.

Тема 8. Інкапсуляція. Частина 2. Приховування, перевантаження і перевизначення. Особливість іменування та властивості конструкторів та властивості деструктора.

Тема 9. Перевантаження. Поняття перевантаження. Перевантаження операторів. Перевантаження функцій у файлі. Перевантаження методів класу. Лямбда-функції.

Тема 10. Регулярні вирази.

## **Семестр 2.**

**Модуль 3.** Поняття наслідування та поліморфізму (7 годин лекцій, 7 годин лабораторних робіт).

Тема 11. Наслідування. Види і типи наслідування. Взаємодія класів (агрегація, використання, асоціація). Вимоги до ієрархії класів.

Тема 12. Поліморфізм. Організація поліморфізму в C ++. Віртуальні функції. Види зв'язування. Абстрактні класи.

Тема 13. Шаблони. Поняття шаблону. Шаблонна функція і шаблонний клас. Особливості компіляції шаблонних функцій і класів. Динамічний і статичний поліморфізм. Поняття дружності і особливості дружності для шаблонів. Види дружності для шаблонів і класів. Шаблони і статичні елементи. Шаблони і нетипові параметри.

**Модуль 4.** Бібліотеки класів C ++(8 годин лекцій, 8 годин лабораторних робіт).

Тема 14. Види бібліотек класів мови C ++. Стандартна бібліотека шаблонів мови C ++ (Standard Template Library, STL).

Тема 15. Контейнери бібліотеки STL (послідовні та асоціативні). Організація роботи з вектором, списком і деком засобами STL. Організація роботи з відображеннями (деревами, множинами) засобами STL.

Тема 16. Групи алгоритмів в STL. Немутаційні алгоритми та мутаційні алгоритми. Алгоритм пошуку find() та алгоритм злиття merge().

Тема 17. Виняткові ситуації в програмі. Можливі помилкові ситуації при роботі C++-програми. Організація обробки виняткової ситуації. Клас виняткових ситуацій в C ++.

**Модуль 5.** Патерни (7 годин лекцій, 7 годин лабораторних робіт).

Тема 18. Поняття патерну. Патерн як ланцюжок відповідальності. Складові патерну. Класифікація патернів. Переваги використання патернів. Приклади.

Тема 19. Породжуючі патерни як засіб утворення об'єктів. Патерн Singleton. Патерн Abstract Factory.

Тема 20. Структурні патерни як засіб утворення класів. Патерн Adapter.

Тема 21. Патерни поведінки як засіб формування поведінки об'єктів. Патерн Strategy. Патерн Template Method.

**Модуль 6.** Технології проектування (8 годин лекцій, 8 годин лабораторних робіт).

Тема 22. Життєвий цикл програми. Види організації життєвого циклу. Етапи проектування ООП-програми. Поняття технології проектування. Архітектура додатку.

Тема 23. Формування варіантів використання програми як можливість виділення функцій програми. Діаграма UML Use case.

Тема 24. Побудова структури ООП-програми. Діаграма класів. Діаграма об'єктів. Діаграма послідовності. Діаграма станів та діаграма дій. Діаграма компонентів та діаграма розміщення.

Тема 25. Додаткові можливості Microsoft Visual Studio Community 2022 для програмування мовою Visual C++.

Дисципліна розрахована на два семестри, 25 лекцій (по 2-3 академічні години кожна та 6 лабораторних робіт по 7-15 академічних годин кожна). Самостійна робота студента розрахована на 180 годин. Кожен семестр завершується екзаменом. В другому семестрі студент виконує курсову роботу за тематикою 5 та 6 модулів.

Лекторка, авторка силабусу та відповідальна за технічну підтримку лабораторних занять – к.т.н., доцент, доцент кафедри інформаційних технологій Петренко Т.Г.

### ***Список посилань (основні джерела)***

1. Stroustrup B. Programming: Principles and Practice Using C++. Addison-Wesley Professional, 2nd edition, 2014, 1312 p.
2. Lippman S. B., Lajoie J., and Moo B. E.. C++ Primer. Addison-Wesley Professional, 5th edition, 2012, 976 p.
3. Meyers S. Effective C++: 55 Specific Ways to Improve Your Programs and Designs. Addison-Wesley Professional, 3rd edition, 2005, 320 p.
4. Booch G. et al. Object-oriented analysis and design with applications. Pearson Education, Inc, 3rd edition, 2007, 717 p.

### ***Список посилань (допоміжні online джерела)***

1. Official Microsoft Documentation. URL: <https://docs.microsoft.com/en-us/cpp/> (Last accessed: 25.08.2024)
2. Visual Studio Magazine. URL: <https://visualstudiomagazine.com/> (Last accessed: 25.08.2024)
3. Stack Overflow. URL: <https://stackoverflow.com/> (Last accessed: 25.08.2024)
4. GitHub and Open-Source Projects. URL: [www.github.com](http://www.github.com) (Last accessed: 25.08.2024)