

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**  
**Кафедра програмування**

**Затверджено**

На засіданні кафедри програмування  
факультету прикладної математики  
Львівського національного університету  
імені Івана Франка  
(протокол № 1 від 29 серпня 2024 р.)



Зав. кафедри к. ф.-м. н., доц. Ярошко С. А.

**Силабус навчальної дисципліни**  
**«Об'єктно-орієнтоване програмування»,**  
**що викладається в межах ОПШ “Прикладна математика”**  
**для здобувачів першого (бакалаврського) рівня вищої освіти**  
**з предметної спеціальності 113 Прикладна математика**  
**галузі знань 11 Математика та статистика**

Львів 2024 р.

|  |  |
|--|--|
| <b>Назва дисципліни</b>  | Об'єктно-орієнтоване програмування   |
| <b>Адреса викладання дисципліни</b>                              | Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000   |
| <b>Факультет та кафедра, за якою закріплена дисципліна</b>       | Факультет прикладної математики та інформатики, кафедра програмування  |
| <b>Галузь знань, шифр та назва спеціальності</b>                 | Галузь знань: 11 Математика та статистика<br>Спеціальність: 113 Прикладна математика   |
| <b>Викладачі дисципліни</b>                                      | Черняхівський Володимир Вікторович, к. ф.-м. н., доц., доцент кафедри програмування  |
| <b>Контактна інформація викладачів</b>                           | Електронна пошта: <a href="mailto:volodymyr.chernyakhivskyy@lnu.edu.ua">volodymyr.chernyakhivskyy@lnu.edu.ua</a><br>веб-сторінка: <a href="https://ami.lnu.edu.ua/employee/cherniakhivskyy">https://ami.lnu.edu.ua/employee/cherniakhivskyy</a>  |
| <b>Консультації з питань навчання по дисципліні відбуваються</b> | Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі онлайн консультації через Zoom чи Microsoft Teams. Для погодження часу онлайн консультацій потрібно писати на електронну пошту викладача.  |
| <b>Сторінка курсу</b>  | <a href="https://ami.lnu.edu.ua/course/object-oriented-programming-system-analysis">https://ami.lnu.edu.ua/course/object-oriented-programming-system-analysis</a>  |
| <b>Інформація про дисципліну</b>                                 | Курс "Об'єктно-орієнтоване програмування" є нормативною дисципліною зі спеціальності 113 Прикладна математика для освітньої програми «Прикладна математика», яку викладають у третьому семестрі в обсязі 4 кредитів (за Європейською кредитно-трансферною системою ECTS).  |
| <b>Коротка анотація дисципліни</b>                               | Викладання дисципліни має на меті сформувати в студентів базову систему знань та навиків в області сучасних прийомів об'єктно-орієнтованого програмування (ООП) та застосування ООП до будови моделей і методів програмування розв'язків задач, а також вивчення мови C++ та прийомів її застосування в обсязі тем курсу. Дисципліна є наступною після курсів "Вступ до програмування" і "Основи програмування", і розглядає різні розділи методів програмування на основі технології ООП: вивчення тематичних засобів алгоритмічної мови C++; аналіз стандартних класів і об'єктів мови C++; типи і структури даних мови C++ та їх зв'язок з технологією ООП; класи і об'єкти, визначені програмно; шаблони класів; успадкування класів; принципи будови бібліотеки шаблонів STL; застосування стандартних класів бібліотеки STL. |
| <b>Мета та цілі дисципліни</b>                                   | Метою нормативної дисципліни «Об'єктно-орієнтоване програмування» є: <ul style="list-style-type: none"> <li>● вивчення алгоритмічної мови C++ в частині об'єктно-орієнтованого програмування в обсязі програми курсу;</li> <li>● вивчення принципів проектування і програмування класів і застосування об'єктів;</li> <li>● отримання навиків проектування і програмування структурних елементів класів;</li> <li>● вивчення принципів успадкування класів і будови ієрархій класів;</li> <li>● вивчення структури і застосування стандартної бібліотеки шаблонів STL;</li> <li>● отримання навиків розробки програм і проєктів на основі технології ООП.</li> </ul>   |
| <b>Література для вивчення дисципліни</b>                        | <i>Основна література</i> <ol style="list-style-type: none"> <li>1. Бублик В.В. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. // К.: ІТ-книга, 2015. – 624 с.</li> <li>2. Мартін Роберт С. Чистий код. Створення, аналіз і рефакторинг: Пер. з англ. / Роберт С. Мартін // К.: Фабула. – 2019. – 368 с. [Електронний ресурс]. – Режим доступу: <a href="https://fabulabook.com/product/chystyj-kod/">https://fabulabook.com/product/chystyj-kod/</a></li> <li>3. w3schools. Підручник C++ [Електронний ресурс]. – Режим доступу: <a href="https://www.w3schools.com/cpp/default.asp">https://www.w3schools.com/cpp/default.asp</a></li> </ol>   |

|                                      |   |
|--------------------------------------|---|
|                                      | <p>4. Документація Microsoft C++, C та Assembler [Електронний ресурс]. – Режим доступу: <a href="https://docs.microsoft.com/uk-ua/cpp/?view=msvc-170&amp;viewFallbackFrom=vs-2019">https://docs.microsoft.com/uk-ua/cpp/?view=msvc-170&amp;viewFallbackFrom=vs-2019</a></p> <p>5. Сайт до вивчення C++ [Електронний ресурс]. – Режим доступу: <a href="https://www.bestprog.net/uk/sitemap_ua/c/">https://www.bestprog.net/uk/sitemap_ua/c/</a></p> <p>6. Ярошко Сергій, Ярошко Оксана. Методи розробки алгоритмів. Програмування мовою C++: навч. посібник / С.А.Ярошко, О.С.Ярошко // Львів: ЛНУ імені Івана Франка. – 2022. – 248с.</p> <p>7. Васильєв Олексій. Програмування на C++ в прикладах і задачах / Олексій Васильєв // Ліра-К. – 2017. – 382с.</p> <p><i>Додаткова література</i></p> <p>8. Ярошко С.А. Методи розробки алгоритмів. Навчальний посібник / С.А. Ярошко // Львів, ЛНУ імені Івана Франка – 2019. [Електронне видання].</p> <p>9. Вікіпедія. C++ [Електронний ресурс]. – Режим доступу: <a href="https://uk.wikipedia.org/wiki/C%2B%2B">https://uk.wikipedia.org/wiki/C%2B%2B</a></p> <p>10. Вікіпедія. Об'єктно-орієнтоване програмування [Електронний ресурс]. – Режим доступу: <a href="https://uk.wikipedia.org/wiki/Об'єктно-орієнтоване_програмування">https://uk.wikipedia.org/wiki/Об'єктно-орієнтоване_програмування</a></p> |
| <b>Обсяг курсу</b>                   | 4 кредити ЄКТС – 120 годин. З них 32 годин лекцій, 32 годин лабораторних занять та 56 годин самостійної роботи.   |
| <b>Очікувані результати навчання</b> | <p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> <li>● алгоритмічну мову C++ разом із засобами об'єктно-орієнтованого програмування в обсязі програми курсу;</li> <li>● принципи проектування і програмування класів і застосування об'єктів;</li> <li>● програмування структурних елементів класів;</li> <li>● принципи успадкування класів і будову ієрархій класів;</li> <li>● проектування шаблонів класів;</li> <li>● структуру і застосування стандартної бібліотеки шаблонів STL.</li> </ul> <p><i>вміти</i></p> <ul style="list-style-type: none"> <li>● 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;</li> <li>● проектувати класи і використовувати технологію об'єктно-орієнтованого програмування в повному обсязі сучасних вимог;</li> <li>● складати шаблони класів для різних типів даних відповідно до задач;</li> <li>● застосовувати стандартні класи бібліотеки шаблонів STL.</li> </ul>   |
| <b>Компетентності</b>                | <p><i>Інтегральна:</i> Здатність застосовувати загальні принципи програмування складних практичних задач засобами мови C++ на професійному рівні. Здатність застосовувати сучасні методи об'єктно-орієнтованого програмування задач різного призначення, проектувати структури даних і алгоритмічні моделі на основі ООП.</p> <p><i>Загальні (ЗК):</i></p> <ul style="list-style-type: none"> <li>● ЗК01. Здатність учитися і оволодівати сучасними знаннями.</li> <li>● ЗК08. Знання та розуміння предметної області та розуміння професійної діяльності.</li> </ul> <p><i>Спеціальні (фахові, предметні) компетентності (ФК):</i></p> <ul style="list-style-type: none"> <li>● ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.</li> <li>● ФК08. Здатність використовувати сучасні технології програмування та тестування програмного забезпечення.</li> <li>● ФК21. Здатність аналізувати можливості наявних технологій розробки програмного забезпечення, обирати та застосовувати найбільш ефективний інструментарій відповідно до специфіки завдання.</li> </ul>  |

| <b>Програмні результати навчання</b> | <ul style="list-style-type: none"> <li>РН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символічних алгоритмів.</li> <li>РН14. Виявляти здатність до самонавчання та продовження професійного розвитку.</li> </ul>  |                                |                          |                              |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
|--------------------------------------|--|--------------------------------|--------------------------|------------------------------|----------------|------------------|---|--|--------|---|--|--|--|--------------------|---|------------------------------|---|---|--------|---|--|--|--|--------------------|---|------------------------------|---|--|--------|---|--|--|--|--------------------|---|------------------------------|---|--|--------|---|--|--|--|--------------------|---|------------------------------|---|--|--------|---|--|--|--|--------------------------------|---|------------------------------|---|--|--------|---|--|--|--|--------------------|---|------------------------------|
| <b>Ключові слова</b>                 | <p>Клас, об'єкт, поле класу, метод класу, конструктор, деструктор, перевантаження оператора, операторна функція, дружня функція, дочірній клас, похідний клас, успадкування, ієрархія класів, віртуальний метод, поліморфізм, перекритий метод, перевантаження метода, чисто віртуальна функція, абстрактний клас, шаблон класу, шаблонний дочірній клас, бібліотека STL, контейнер STL, ітератор STL, алгоритм STL, список STL, стек, черга, послідовний контейнер, асоціативний контейнер, контейнер словника, ключ, функтор, функціональний об'єкт, предикат, виключна ситуація, проект ООП, модель включення класів.</p>   |                                |                          |                              |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
| <b>Формат курсу</b>                  | <p>Очний: проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах форсмажорних обставин – онлайнвий на платформі Microsoft Teams.</p>   |                                |                          |                              |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
| <b>Теми</b>                          | <table border="1"> <thead> <tr> <th data-bbox="491 831 568 887">Тиж-день</th> <th data-bbox="568 831 1099 887">Тема, план, короткі тези</th> <th data-bbox="1099 831 1246 887">Форма заняття</th> <th data-bbox="1246 831 1350 887">Тривалість год</th> <th data-bbox="1350 831 1535 887">Термін виконання</th> </tr> </thead> <tbody> <tr> <td data-bbox="491 887 568 1133">1</td> <td data-bbox="568 887 1099 1133"> <p>Класи і об'єкти. Будова класу. Зміст класів і об'єктів. Схема визначення класів в С++. Проектування класу. Ключові слова, поля, методи, спосіб доступу. Програмна реалізація класу. Перевантаження методів класу. Застосування класу. Об'єкти. Модель організації файлів класу.</p> <p>1)Задачі на повторення базової частини мови С++. Проектування, програмування, налагодження, тестування. 2)Задачі і програмна реалізація прикладів за змістом теми 1.</p> </td> <td data-bbox="1099 887 1246 1032">Лекція</td> <td data-bbox="1246 887 1350 1032">2</td> <td data-bbox="1350 887 1535 1032"></td> </tr> <tr> <td data-bbox="491 1032 568 1133"></td> <td data-bbox="568 1032 1099 1133"></td> <td data-bbox="1099 1032 1246 1133">Лабораторна робота</td> <td data-bbox="1246 1032 1350 1133">2</td> <td data-bbox="1350 1032 1535 1133">Наступне лабораторне заняття</td> </tr> <tr> <td data-bbox="491 1133 568 1301">2</td> <td data-bbox="568 1133 1099 1301"> <p>Конструктори і деструктори класів. Зміст і будова конструктора. Правила будови конструкторів. Правило замовчування. Перевантаження конструкторів. Конструктор копіювання. Деструктори.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 2.</p> </td> <td data-bbox="1099 1133 1246 1301">Лекція</td> <td data-bbox="1246 1133 1350 1301">2</td> <td data-bbox="1350 1133 1535 1301"></td> </tr> <tr> <td data-bbox="491 1301 568 1424"></td> <td data-bbox="568 1301 1099 1424"></td> <td data-bbox="1099 1301 1246 1424">Лабораторна робота</td> <td data-bbox="1246 1301 1350 1424">2</td> <td data-bbox="1350 1301 1535 1424">Наступне лабораторне заняття</td> </tr> <tr> <td data-bbox="491 1424 568 1503">3</td> <td data-bbox="568 1424 1099 1503"> <p>Динамічні об'єкти і вказівники. Масиви об'єктів. Вказівники на об'єкти. Посилання на існуючі об'єкти. Динамічне створення об'єктів. Вказівники на об'єкти як параметри функцій. Динамічні об'єкти як результати методів. Вказівник на власний об'єкт.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 3.</p> </td> <td data-bbox="1099 1424 1246 1503">Лекція</td> <td data-bbox="1246 1424 1350 1503">2</td> <td data-bbox="1350 1424 1535 1503"></td> </tr> <tr> <td data-bbox="491 1503 568 1671"></td> <td data-bbox="568 1503 1099 1671"></td> <td data-bbox="1099 1503 1246 1671">Лабораторна робота</td> <td data-bbox="1246 1503 1350 1671">2</td> <td data-bbox="1350 1503 1535 1671">Наступне лабораторне заняття</td> </tr> <tr> <td data-bbox="491 1671 568 1749">4</td> <td data-bbox="568 1671 1099 1749"> <p>Перевантаження операторів. Операторні функції. Зміст перевантаження операторів. Приклад перевантаження операції. Аналіз визначення і результатів. Повторне перевантаження операцій. Перевантаження комбінованих операторів присвоєння. Перевантаження операції присвоєння. Перевантаження операцій логічного типу результату.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 4.</p> </td> <td data-bbox="1099 1671 1246 1749">Лекція</td> <td data-bbox="1246 1671 1350 1749">2</td> <td data-bbox="1350 1671 1535 1749"></td> </tr> <tr> <td data-bbox="491 1749 568 1872"></td> <td data-bbox="568 1749 1099 1872"></td> <td data-bbox="1099 1749 1246 1872">Лабораторна робота</td> <td data-bbox="1246 1749 1350 1872">2</td> <td data-bbox="1350 1749 1535 1872">Наступне лабораторне заняття</td> </tr> <tr> <td data-bbox="491 1872 568 1951">5</td> <td data-bbox="568 1872 1099 1951"> <p>Перевантаження спеціальних операторів. Префіксна і постфіксна форма операторів інкременту і декременту. Перевантаження оператора індексування []. Перевантаження оператора круглих дужок (). Дружні функції.</p> <p>1)Задачі і програмна реалізація прикладів за змістом теми 5. 2)Контрольна робота за темами 1-5.</p> </td> <td data-bbox="1099 1872 1246 1951">Лекція</td> <td data-bbox="1246 1872 1350 1951">2</td> <td data-bbox="1350 1872 1535 1951"></td> </tr> <tr> <td data-bbox="491 1951 568 2051"></td> <td data-bbox="568 1951 1099 2051"></td> <td data-bbox="1099 1951 1246 2051">Лабораторна робота; контрольна</td> <td data-bbox="1246 1951 1350 2051">2</td> <td data-bbox="1350 1951 1535 2051">Наступне лабораторне заняття</td> </tr> <tr> <td data-bbox="491 2051 568 2121">6</td> <td data-bbox="568 2051 1099 2121"> <p>Дочірні класи. Загальна будова дочірнього класу. Базові засади успадкування. Проектування дочірнього класу. Аналіз будови дочірнього класу. Ієрархія класів. Правило присвоєння об'єктів.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 6.</p> </td> <td data-bbox="1099 2051 1246 2121">Лекція</td> <td data-bbox="1246 2051 1350 2121">2</td> <td data-bbox="1350 2051 1535 2121"></td> </tr> <tr> <td data-bbox="491 2121 568 2121"></td> <td data-bbox="568 2121 1099 2121"></td> <td data-bbox="1099 2121 1246 2121">Лабораторна робота</td> <td data-bbox="1246 2121 1350 2121">2</td> <td data-bbox="1350 2121 1535 2121">Наступне лабораторне заняття</td> </tr> </tbody> </table> | Тиж-день                       | Тема, план, короткі тези | Форма заняття                | Тривалість год | Термін виконання | 1 | <p>Класи і об'єкти. Будова класу. Зміст класів і об'єктів. Схема визначення класів в С++. Проектування класу. Ключові слова, поля, методи, спосіб доступу. Програмна реалізація класу. Перевантаження методів класу. Застосування класу. Об'єкти. Модель організації файлів класу.</p> <p>1)Задачі на повторення базової частини мови С++. Проектування, програмування, налагодження, тестування. 2)Задачі і програмна реалізація прикладів за змістом теми 1.</p> | Лекція | 2 |  |  |  | Лабораторна робота | 2 | Наступне лабораторне заняття | 2 | <p>Конструктори і деструктори класів. Зміст і будова конструктора. Правила будови конструкторів. Правило замовчування. Перевантаження конструкторів. Конструктор копіювання. Деструктори.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 2.</p> | Лекція | 2 |  |  |  | Лабораторна робота | 2 | Наступне лабораторне заняття | 3 | <p>Динамічні об'єкти і вказівники. Масиви об'єктів. Вказівники на об'єкти. Посилання на існуючі об'єкти. Динамічне створення об'єктів. Вказівники на об'єкти як параметри функцій. Динамічні об'єкти як результати методів. Вказівник на власний об'єкт.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 3.</p> | Лекція | 2 |  |  |  | Лабораторна робота | 2 | Наступне лабораторне заняття | 4 | <p>Перевантаження операторів. Операторні функції. Зміст перевантаження операторів. Приклад перевантаження операції. Аналіз визначення і результатів. Повторне перевантаження операцій. Перевантаження комбінованих операторів присвоєння. Перевантаження операції присвоєння. Перевантаження операцій логічного типу результату.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 4.</p> | Лекція | 2 |  |  |  | Лабораторна робота | 2 | Наступне лабораторне заняття | 5 | <p>Перевантаження спеціальних операторів. Префіксна і постфіксна форма операторів інкременту і декременту. Перевантаження оператора індексування []. Перевантаження оператора круглих дужок (). Дружні функції.</p> <p>1)Задачі і програмна реалізація прикладів за змістом теми 5. 2)Контрольна робота за темами 1-5.</p> | Лекція | 2 |  |  |  | Лабораторна робота; контрольна | 2 | Наступне лабораторне заняття | 6 | <p>Дочірні класи. Загальна будова дочірнього класу. Базові засади успадкування. Проектування дочірнього класу. Аналіз будови дочірнього класу. Ієрархія класів. Правило присвоєння об'єктів.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 6.</p> | Лекція | 2 |  |  |  | Лабораторна робота | 2 | Наступне лабораторне заняття |
| Тиж-день                             | Тема, план, короткі тези   | Форма заняття                  | Тривалість год           | Термін виконання             |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
| 1                                    | <p>Класи і об'єкти. Будова класу. Зміст класів і об'єктів. Схема визначення класів в С++. Проектування класу. Ключові слова, поля, методи, спосіб доступу. Програмна реалізація класу. Перевантаження методів класу. Застосування класу. Об'єкти. Модель організації файлів класу.</p> <p>1)Задачі на повторення базової частини мови С++. Проектування, програмування, налагодження, тестування. 2)Задачі і програмна реалізація прикладів за змістом теми 1.</p>   | Лекція                         | 2                        |                              |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
|                                      |  | Лабораторна робота             | 2                        | Наступне лабораторне заняття |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
| 2                                    | <p>Конструктори і деструктори класів. Зміст і будова конструктора. Правила будови конструкторів. Правило замовчування. Перевантаження конструкторів. Конструктор копіювання. Деструктори.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 2.</p>  | Лекція                         | 2                        |                              |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
|                                      |  | Лабораторна робота             | 2                        | Наступне лабораторне заняття |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
| 3                                    | <p>Динамічні об'єкти і вказівники. Масиви об'єктів. Вказівники на об'єкти. Посилання на існуючі об'єкти. Динамічне створення об'єктів. Вказівники на об'єкти як параметри функцій. Динамічні об'єкти як результати методів. Вказівник на власний об'єкт.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 3.</p>   | Лекція                         | 2                        |                              |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
|                                      |  | Лабораторна робота             | 2                        | Наступне лабораторне заняття |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
| 4                                    | <p>Перевантаження операторів. Операторні функції. Зміст перевантаження операторів. Приклад перевантаження операції. Аналіз визначення і результатів. Повторне перевантаження операцій. Перевантаження комбінованих операторів присвоєння. Перевантаження операції присвоєння. Перевантаження операцій логічного типу результату.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 4.</p>   | Лекція                         | 2                        |                              |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
|                                      |  | Лабораторна робота             | 2                        | Наступне лабораторне заняття |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
| 5                                    | <p>Перевантаження спеціальних операторів. Префіксна і постфіксна форма операторів інкременту і декременту. Перевантаження оператора індексування []. Перевантаження оператора круглих дужок (). Дружні функції.</p> <p>1)Задачі і програмна реалізація прикладів за змістом теми 5. 2)Контрольна робота за темами 1-5.</p>   | Лекція                         | 2                        |                              |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
|                                      |  | Лабораторна робота; контрольна | 2                        | Наступне лабораторне заняття |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
| 6                                    | <p>Дочірні класи. Загальна будова дочірнього класу. Базові засади успадкування. Проектування дочірнього класу. Аналіз будови дочірнього класу. Ієрархія класів. Правило присвоєння об'єктів.</p> <p>Задачі і програмна реалізація прикладів за змістом теми 6.</p>   | Лекція                         | 2                        |                              |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |
|                                      |  | Лабораторна робота             | 2                        | Наступне лабораторне заняття |                |                  |   |  |        |   |  |  |  |                    |   |                              |   |   |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                    |   |                              |   |  |        |   |  |  |  |                                |   |                              |   |  |        |   |  |  |  |                    |   |                              |

|    |   |                                |   |                              |
|----|---|--------------------------------|---|------------------------------|
| 7  | Віртуальні методи і поліморфізм. Деякі проблеми перекриття методів. Правила віртуальності. Різниця між перекриттям і перевантаженням віртуальних методів. Чисто віртуальні функції і абстрактні класи. Приклад програмної реалізації абстрактного класу.  | Лекція                         | 2 |                              |
|    | Задачі і програмна реалізація прикладів за змістом теми 7.  | Лабораторна робота             | 2 | Наступне лабораторне заняття |
| 8  | Шаблони класів. Загальні принципи будови шаблонів. Будова шаблону класу. Правила будови. Об'єкти на основі шаблону. Додаткові способи визначення шаблону класу. Дочірні класи на базі шаблону. Особливості шаблону для нестандартних типів даних.   | Лекція                         | 2 |                              |
|    | 1)Задачі і програмна реалізація прикладів за змістом теми 8. 2)Контрольна робота за темами 6-8.   | Лабораторна робота; контрольна | 2 | Наступне лабораторне заняття |
| 9  | Стандартний клас string. Способи опрацювання текстових рядків в C++. Конструктори і оператори класу string. Особливості виконання операторів класу string. Характеристика методів класу string. Приклади задач для string.  | Лекція                         | 2 |                              |
|    | Задачі і програмна реалізація прикладів за змістом теми 9.  | Лабораторна робота             | 2 | Наступне лабораторне заняття |
| 10 | Принципи будови стандартної бібліотеки шаблонів. Огляд бібліотеки STL. Контейнер, ітератор, алгоритми. Діапазон. Загальна характеристика списків. Категорії операцій над списками. Комплексний приклад задачі застосування списку і програмна реалізація.   | Лекція                         | 2 |                              |
|    | Задачі і програмна реалізація прикладів за змістом теми 10.   | Лабораторна робота             | 2 | Наступне лабораторне заняття |
| 11 | Стеки і черги стандартної бібліотеки шаблонів. Принцип функціонування стека. Клас стека бібліотеки STL, методи. Приклади задач на використання стека. Принцип функціонування черги. Клас черги бібліотеки STL, методи. Комплексний приклад задачі на використання черги і програмна реалізація.           | Лекція                         | 2 |                              |
|    | Задачі і програмна реалізація прикладів за змістом теми 11.   | Лабораторна робота             | 2 | Наступне лабораторне заняття |
| 12 | Асоціативні контейнери STL. Контейнер словника map. Різниця між послідовними і асоціативними контейнерами. Характеристика класу map. Об'єкти для порівняння ключів. Загальна характеристика методів класу map. Комплексний приклад задачі на використання словників і програмна реалізація.               | Лекція                         | 2 |                              |
|    | Задачі і програмна реалізація прикладів за змістом теми 12.   | Лабораторна робота             | 2 | Наступне лабораторне заняття |
| 13 | Функціональні об'єкти (функтори). Зміст і види функціональних об'єктів. Предикати. Приклад задачі і програмної реалізації на використання словника map і функторів. Бінарні предикати. Приклад програмної реалізації з використанням бінарних предикатів.   | Лекція                         | 2 |                              |
|    | Задачі і програмна реалізація прикладів за змістом теми 13.   | Лабораторна робота             | 2 | Наступне лабораторне заняття |
| 14 | Виключні ситуації на основі класів. Принципи опрацювання виключних ситуацій. Організація програм для захисту від помилок. Проектування класів для опрацювання виключних ситуацій. Приклад застосування класу. Приклад архітектури і реалізації програми з використанням класів виключних ситуацій.        | Лекція                         | 2 |                              |
|    | 1)Задачі і програмна реалізація прикладів за змістом теми 14. 2)Контрольна робота за темами 9-14.   | Лабораторна робота; контрольна | 2 | Наступне лабораторне заняття |
| 15 | Методи проектування програм на основі ООП. Проектування полів даних класів. Проектування архітектури, масив об'єктів, об'єкт масиву. Проектування відношень успадкування. Приклад задачі і програми на відношення успадкування. Співвідношення класів і структур мови C++. Модель включення класів в C++. | Лекція                         | 2 |                              |
|    | Задачі і програмна реалізація прикладів за змістом теми 15.   | Лабораторна робота             | 2 | Наступне лабораторне заняття |
| 16 | Підсумковий огляд засобів ООП мови C++, порівняння з засобами інших мов. Роль ООП в розробці сучасних програмних проектів. Вимоги до володіння методами практичного програмування на основі технології ООП.   | Лекція                         | 2 |                              |

|   |  |                    |   |               |
|---|--|--------------------|---|---------------|
|   | Додаткові задачі за темами курсу. Підготовка до іспиту.  | Лабораторна робота | 2 | Початок сесії |
| <b>Підсумковий контроль, форма</b>  | Іспит в кінці семестру   |                    |   |               |
| <b>Пререквізити</b>   | Для вивчення курсу студенти потребують базових знань з дисциплін: вступ до програмування, основи програмування, алгоритми обчислювальних процесів. Одночасно з вивченням програмування студенти проходять навчальну обчислювальну практику, впродовж якої виконують завдання з програмування, поглиблюють набуті в курсі знання та удосконалюють навички.  |                    |   |               |
| <b>Навчальні методи та техніки, які використовують під час викладання курсу</b> | Лекції з мультимедійними презентаціями та з демонстрацією прийомів програмування засобами мови C++; лабораторні заняття у вигляді проектування алгоритмів і програм, виконання практичних завдань та проектів; індивідуальні домашні завдання на програмування розв'язків задач; самостійне опрацювання навчальних матеріалів: підручників, конспектів лекцій, електронних ресурсів, готових програм мовою C++, додаткових матеріалів, розміщених у хмарному сховищі (Microsoft Teams, Google Диск). Обговорення теоретичного та практичного матеріалу в онлайн сервісах, формулювання творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного і практичного матеріалу.  |                    |   |               |
| <b>Необхідне обладнання</b>   | Для проведення лекцій: комп'ютер, проектор, доступ до мережі інтернет.<br>Для проведення лабораторних та виконання завдань: комп'ютер; ОС Windows/Linux; доступ до інтернету; середовища програмування мовою C++ (Microsoft Visual Studio, Code Blocks тощо).<br>Уся література і робочі матеріали, які студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.  |                    |   |               |
| <b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>      | <p><b>Оцінювання</b> проводиться за 100-бальною шкалою. 50 балів нараховують за виконання лабораторних завдань, домашніх завдань, контрольних робіт впродовж семестру, ще 50 балів – за виконання екзаменаційного завдання. Лабораторні завдання індивідуальні. Домашні завдання можуть бути індивідуальні та командні. Упродовж семестру студент виконує не менше 10 лабораторних і домашніх, які оцінюють різною шкалою залежно від складності (критерії надаються студентам до кожного лабораторного чи домашнього завдання). Для кожного завдання визначено термін виконання – зазвичай до наступного лабораторного заняття. Вчасно виконані завдання <b>поточного контролю</b> оцінюють так (у відсотках від максимальної оцінки):</p> <ul style="list-style-type: none"> <li>• 100% – умови завдання виконано повністю, алгоритми складено правильно, програма містить належні коментарі, роботу програми перевірено на достатньому наборі тестових даних, автор відповідає на всі запитання щодо використаних підходів, чітко інтерпретує отримані результати, немає ознак недоброчесності;</li> <li>• 80% – наведено логічно правильну послідовність розв'язування, алгоритми складено правильно, бракує окремих коментарів чи тестів, автор не досить повно пояснює використані підходи, немає ознак недоброчесності;</li> <li>• 60% – у правильній послідовності розв'язування допущено окремі помилки, які автор уміє виправити після зауваження викладача, бракує коментарів чи тестів, на запитання щодо використаних підходів автор відповідає з помилками, немає ознак недоброчесності;</li> <li>• 40% – у правильній послідовності розв'язування пропущено окремі етапи, завдання виконано частково, автор не розуміє недоліків поданої роботи, не вміє їх виправити, немає ознак недоброчесності;</li> <li>• 20% – завдання виконано частково, немає тестів, програма працює правильно для окремих наборів вхідних даних, автор не може самостійно інтерпретувати отримані результати, виправити помилки, немає ознак недоброчесності;</li> <li>• 0% – завдання не виконано, написана програма не відповідає умові, або ж виявлено ознаки недоброчесності: запозичення, фрагменти коду, дію яких</li> </ul> |                    |   |               |

|                   |  |
|-------------------|--|
|                   | <p>автор пояснити не може, автор не володіє відповідним теоретичним матеріалом тощо;</p> <ul style="list-style-type: none"> <li>• можуть бути нараховані додаткові бали за повністю виконане завдання, яке містить кілька способів розв'язування, використовує особливо ефективний спосіб, демонструє креативність автора тощо.</li> </ul> <p>Запізнення терміну виконання завдання зменшує максимальну оцінку за завдання, ступінь зменшення визначає викладач до кожної роботи.</p> <p>На <b>екзамен</b> студент отримує одну комплексну задачу, для якої треба програмно реалізувати низку вимог за темами прочитаного курсу:</p> <ul style="list-style-type: none"> <li>– побудувати абстрактний клас;</li> <li>– побудувати батьківський клас;</li> <li>– побудувати дочірній клас;</li> <li>– конструктори і деструктор;</li> <li>– методи редагування змісту об'єкта;</li> <li>– методи відображення об'єкта в текстовий файл і на екран;</li> <li>– методи доступу до полів об'єкта, методи обчислювальних операцій;</li> <li>– перевантаження операцій бінарних і унарних;</li> <li>– використання стандартної бібліотеки шаблонів STL.</li> </ul> <p>Кожний пункт умови реалізації задачі зараховується лише за наявності відповідних тестів, які демонструють його виконання. Побудувати і обґрунтувати тести студент повинний самостійно, вони є частиною оцінки.</p> <p>Повний перелік вимог, шкала оцінювання і зразок завдання будуть надані студентам до закінчення лекційних і лабораторних занять.</p> <p>Оцінка за екзаменаційне завдання може отримати додаткові бали, якщо завдання чи проєкти студента, виконані впродовж семестру, мали закінчений характер і високий рівень фахової реалізації.</p> <p><b>Відвідання занять</b> є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Активність під час проведення лекцій і лабораторних заохочується балами при оцінюванні відповідного лабораторного завдання чи проєкта. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів робіт, передбачених курсом. Виконані роботи завантажують у відповідне хмарне сховище. Альтернативою відвідування лабораторних занять в університеті може бути дистанційна онлайн робота за розкладом проведення занять.</p> <p><b>Академічна доброчесність:</b> очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп'ютерних програм як своїх становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p> |
| <b>Опитування</b> | Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.  |