

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 2023 р.)



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

Силабус навчальної дисципліни
«Об'єктно-орієнтоване програмування»,
що викладається в межах першого (бакалаврського) рівня
вищої освіти для здобувачів
з спеціальності 113 Прикладна математика

Львів 2023 р.

Назва дисципліни	Об'єктно-орієнтоване програмування
Адреса викладання дисципліни	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики, кафедра програмування
Галузь знань, шифр та назва спеціальності	Галузь знань: 11 Математика та статистика Спеціальність: 113 Прикладна математика
Викладачі дисципліни	Черняхівський Володимир Вікторович, к. ф.-м. н., доц., доцент кафедри програмування
Контактна інформація викладачів	Електронна пошта: volodymyr.chernyakhivskyy@lnu.edu.ua веб-сторінка: https://ami.lnu.edu.ua/employee/cherniakhivskiy
Консультації з питань навчання по дисципліні відбуваються	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.
Сторінка курсу	https://ami.lnu.edu.ua/course/object-oriented-programming-system-analysis
Інформація про дисципліну	Курс «Об'єктно-орієнтоване програмування» є нормативною дисципліною зі спеціальності 113 Прикладна математика, яку викладають у третьому семестрі в обсязі 4 кредитів (за Європейською кредитно-трансферною системою ECTS).
Коротка анотація дисципліни	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.
Мета та цілі дисципліни	Метою нормативної дисципліни «Об'єктно-орієнтоване програмування» є: <ul style="list-style-type: none"> ● вивчення алгоритмічної мови C++ в частині об'єктно-орієнтованого програмування в обсязі програми курсу; ● вивчення принципів проектування і програмування класів і застосування об'єктів; ● отримання навиків проектування і програмування структурних елементів класів; ● вивчення принципів успадкування класів і будови ієрархій класів; ● вивчення структури і застосування стандартної бібліотеки шаблонів STL; ● отримання навиків розробки програм і проектів на основі технології ООП.
Література для вивчення дисципліни	<i>Основна література</i> <ol style="list-style-type: none"> 1. Бублик В.В. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. // К.: ІТ-книга, 2015. – 624 с. 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: https://fabulabook.com/product/chystyj-kod/

	<ol style="list-style-type: none"> 3. w3schools. Підручник C++ [Електронний ресурс]. – Режим доступу: https://www.w3schools.com/cpp/default.asp 4. Документація Microsoft C++, C та Assembler [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/uk-ua/cpp/?view=msvc-170&viewFallbackFrom=vs-2019 5. Сайт до вивчення C++ [Електронний ресурс]. – Режим доступу: https://www.bestprog.net/uk/sitemap_ua/c/ 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. 7. Васильєв Олексій. Програмування на C++ в прикладах і задачах / Олексій Васильєв // Ліра-К. – 2017. – 382с. <p><i>Додаткова література</i></p> <ol style="list-style-type: none"> 8. Ярошко С.А. Методи розробки алгоритмів. Навчальний посібник / С.А. Ярошко // Львів, ЛНУ імені Івана Франка – 2019. [Електронне видання]. 9. Вікіпедія. C++ [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/C%2B%2B 10. Вікіпедія. Об'єктно-орієнтоване програмування [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Об%27єктно-орієнтоване_програмування
Обсяг курсу	4 кредити ЄКТС – 120 годин. З них 32 годин лекцій, 32 годин лабораторних занять та 56 годин самостійної роботи.
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> ● алгоритмічну мову C++ разом із засобами об'єктно-орієнтованого програмування в обсязі програми курсу; ● принципи проектування і програмування класів і застосування об'єктів; ● програмування структурних елементів класів; ● принципи успадкування класів і будову ієрархій класів; ● проектування шаблонів класів; ● структуру і застосування стандартної бібліотеки шаблонів STL. <p><i>вміти</i></p> <ul style="list-style-type: none"> ● проектувати, кодувати, тестувати і налагоджувати програми C++ в обсязі всіх можливостей мови, будувати програмну взаємодію технології об'єктно-орієнтованого програмування з іншими методами і парадигмами; ● проектувати класи і використовувати технологію об'єктно-орієнтованого програмування в повному обсязі сучасних вимог; ● складати шаблони класів для різних типів даних відповідно до задач; ● застосовувати стандартні класи бібліотеки шаблонів STL. <p>Курс забезпечує набуття таких компетентностей та програмних результатів навчання:</p> <p>Фахові компетентності спеціальності:</p> <p>ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.</p> <p>Програмні результати навчання:</p> <p>РН09. Будувати ефективні щодо точності обчислень, стійкості, швидкодії та витрат системних ресурсів алгоритми для чисельного дослідження математичних моделей та розв'язання практичних задач.</p> <p>РН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символічних алгоритмів.</p>
Ключові слова	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.																																																																																	
Формат курсу	Очний: проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах форсмажорних обставин – онлайнний на платформі Microsoft Teams.																																																																																	
Теми	<table border="1"> <thead> <tr> <th>Тиж-день</th> <th>Тема, план, короткі тези</th> <th>Форма заняття</th> <th>Тривалість год</th> <th>Термін виконання</th> </tr> </thead> <tbody> <tr> <td rowspan="2">1</td> <td>Класи і об'єкти. Будова класу. Зміст класів і об'єктів. Схема визначення класів в C++. Проектування класу. Ключові слова, поля, методи, спосіб доступу. Програмна реалізація класу. Перевантаження методів класу. Застосування класу. Об'єкти. Модель організації файлів класу.</td> <td>Лекція, Самостійна робота</td> <td>2 3,5</td> <td></td> </tr> <tr> <td>1)Задачі на повторення базової частини мови C++. Проектування, програмування, налагодження, тестування. 2)Задачі і програмна реалізація прикладів за змістом теми 1.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">2</td> <td>Конструктори і деструктори класів. Зміст і будова конструктора. Правила будови конструкторів. Правило замовчування. Перевантаження конструкторів. Конструктор копіювання. Деструктори.</td> <td>Лекція, Самостійна робота</td> <td>2 3,5</td> <td></td> </tr> <tr> <td>Задачі і програмна реалізація прикладів за змістом теми 2.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">3</td> <td>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.</td> <td>Лекція, Самостійна робота</td> <td>2 3,5</td> <td></td> </tr> <tr> <td>Задачі і програмна реалізація прикладів за змістом теми 3.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">4</td> <td>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.</td> <td>Лекція, Самостійна робота</td> <td>2 3,5</td> <td></td> </tr> <tr> <td>Задачі і програмна реалізація прикладів за змістом теми 4.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">5</td> <td>Перевантаження спеціальних операторів. Префіксна і постфіксна форма операторів інкременту і декременту. Перевантаження оператора індексування []. Перевантаження оператора круглих дужок (). Дружні функції.</td> <td>Лекція, Самостійна робота</td> <td>2 3,5</td> <td></td> </tr> <tr> <td>1)Задачі і програмна реалізація прикладів за змістом теми 5. 2)Контрольна робота за темами 1-5.</td> <td>Лабораторна робота; контрольна</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">6</td> <td>Дочірні класи. Загальна будова дочірнього класу. Базові засади успадкування. Проектування дочірнього класу. Аналіз будови дочірнього класу. Ієрархія класів. Правило присвоєння об'єктів.</td> <td>Лекція, Самостійна робота</td> <td>2 3,5</td> <td></td> </tr> <tr> <td>Задачі і програмна реалізація прикладів за змістом теми 6.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">7</td> <td>Віртуальні методи і поліморфізм. Деякі проблеми перекритих методів. Правила віртуальності. Різниця між перекриттям і перевантаженням віртуальних методів. Чисто віртуальні функції і абстрактні класи. Приклад програмної реалізації абстрактного класу.</td> <td>Лекція, Самостійна робота</td> <td>2 3,5</td> <td></td> </tr> <tr> <td>Задачі і програмна реалізація прикладів за змістом теми 7.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">8</td> <td>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.</td> <td>Лекція, Самостійна робота</td> <td>2 3,5</td> <td></td> </tr> <tr> <td>1)Задачі і програмна реалізація прикладів за змістом</td> <td>Лабораторна</td> <td>2</td> <td>Наступне</td> </tr> </tbody> </table>					Тиж-день	Тема, план, короткі тези	Форма заняття	Тривалість год	Термін виконання	1	Класи і об'єкти. Будова класу. Зміст класів і об'єктів. Схема визначення класів в C++. Проектування класу. Ключові слова, поля, методи, спосіб доступу. Програмна реалізація класу. Перевантаження методів класу. Застосування класу. Об'єкти. Модель організації файлів класу.	Лекція, Самостійна робота	2 3,5		1)Задачі на повторення базової частини мови C++. Проектування, програмування, налагодження, тестування. 2)Задачі і програмна реалізація прикладів за змістом теми 1.	Лабораторна робота	2	Наступне лабораторне заняття	2	Конструктори і деструктори класів. Зміст і будова конструктора. Правила будови конструкторів. Правило замовчування. Перевантаження конструкторів. Конструктор копіювання. Деструктори.	Лекція, Самостійна робота	2 3,5		Задачі і програмна реалізація прикладів за змістом теми 2.	Лабораторна робота	2	Наступне лабораторне заняття	3	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.	Лекція, Самостійна робота	2 3,5		Задачі і програмна реалізація прикладів за змістом теми 3.	Лабораторна робота	2	Наступне лабораторне заняття	4	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.	Лекція, Самостійна робота	2 3,5		Задачі і програмна реалізація прикладів за змістом теми 4.	Лабораторна робота	2	Наступне лабораторне заняття	5	Перевантаження спеціальних операторів. Префіксна і постфіксна форма операторів інкременту і декременту. Перевантаження оператора індексування []. Перевантаження оператора круглих дужок (). Дружні функції.	Лекція, Самостійна робота	2 3,5		1)Задачі і програмна реалізація прикладів за змістом теми 5. 2)Контрольна робота за темами 1-5.	Лабораторна робота; контрольна	2	Наступне лабораторне заняття	6	Дочірні класи. Загальна будова дочірнього класу. Базові засади успадкування. Проектування дочірнього класу. Аналіз будови дочірнього класу. Ієрархія класів. Правило присвоєння об'єктів.	Лекція, Самостійна робота	2 3,5		Задачі і програмна реалізація прикладів за змістом теми 6.	Лабораторна робота	2	Наступне лабораторне заняття	7	Віртуальні методи і поліморфізм. Деякі проблеми перекритих методів. Правила віртуальності. Різниця між перекриттям і перевантаженням віртуальних методів. Чисто віртуальні функції і абстрактні класи. Приклад програмної реалізації абстрактного класу.	Лекція, Самостійна робота	2 3,5		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, Самостійна робота	2 3,5		1)Задачі і програмна реалізація прикладів за змістом	Лабораторна	2	Наступне
Тиж-день	Тема, план, короткі тези	Форма заняття	Тривалість год	Термін виконання																																																																														
1	Класи і об'єкти. Будова класу. Зміст класів і об'єктів. Схема визначення класів в C++. Проектування класу. Ключові слова, поля, методи, спосіб доступу. Програмна реалізація класу. Перевантаження методів класу. Застосування класу. Об'єкти. Модель організації файлів класу.	Лекція, Самостійна робота	2 3,5																																																																															
	1)Задачі на повторення базової частини мови C++. Проектування, програмування, налагодження, тестування. 2)Задачі і програмна реалізація прикладів за змістом теми 1.	Лабораторна робота	2	Наступне лабораторне заняття																																																																														
2	Конструктори і деструктори класів. Зміст і будова конструктора. Правила будови конструкторів. Правило замовчування. Перевантаження конструкторів. Конструктор копіювання. Деструктори.	Лекція, Самостійна робота	2 3,5																																																																															
	Задачі і програмна реалізація прикладів за змістом теми 2.	Лабораторна робота	2	Наступне лабораторне заняття																																																																														
3	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.	Лекція, Самостійна робота	2 3,5																																																																															
	Задачі і програмна реалізація прикладів за змістом теми 3.	Лабораторна робота	2	Наступне лабораторне заняття																																																																														
4	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.	Лекція, Самостійна робота	2 3,5																																																																															
	Задачі і програмна реалізація прикладів за змістом теми 4.	Лабораторна робота	2	Наступне лабораторне заняття																																																																														
5	Перевантаження спеціальних операторів. Префіксна і постфіксна форма операторів інкременту і декременту. Перевантаження оператора індексування []. Перевантаження оператора круглих дужок (). Дружні функції.	Лекція, Самостійна робота	2 3,5																																																																															
	1)Задачі і програмна реалізація прикладів за змістом теми 5. 2)Контрольна робота за темами 1-5.	Лабораторна робота; контрольна	2	Наступне лабораторне заняття																																																																														
6	Дочірні класи. Загальна будова дочірнього класу. Базові засади успадкування. Проектування дочірнього класу. Аналіз будови дочірнього класу. Ієрархія класів. Правило присвоєння об'єктів.	Лекція, Самостійна робота	2 3,5																																																																															
	Задачі і програмна реалізація прикладів за змістом теми 6.	Лабораторна робота	2	Наступне лабораторне заняття																																																																														
7	Віртуальні методи і поліморфізм. Деякі проблеми перекритих методів. Правила віртуальності. Різниця між перекриттям і перевантаженням віртуальних методів. Чисто віртуальні функції і абстрактні класи. Приклад програмної реалізації абстрактного класу.	Лекція, Самостійна робота	2 3,5																																																																															
	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, Самостійна робота	2 3,5																																																																															
	1)Задачі і програмна реалізація прикладів за змістом	Лабораторна	2	Наступне																																																																														

		теми 8. 2)Контрольна робота за темами 6-8.	робота; контрольна		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, Самостійна робота	2 3,5		
	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, Самостійна робота	2 3,5		
	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, Самостійна робота	2 3,5		
	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, Самостійна робота	2 3,5		
	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, Самостійна робота	2 3,5		
	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, Самостійна робота	2 3,5		
	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, Самостійна робота	2 3,5		
	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 проєктів. Вимоги до володіння методами практичного програмування на основі технології ООП.	Лекція, Самостійна робота	2 3,5		
	Додаткові задачі за темами курсу. Підготовка до іспиту.	Лабораторна робота	2		Початок сесії
Підсумковий контроль, форма	Екзамен				
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін: вступ до програмування, основи програмування, алгоритми обчислювальних процесів.				
Навчальні методи та техніки, які	Лекції з мультимедійними презентаціями та з демонстрацією прийомів програмування засобами мови C++; лабораторні заняття у вигляді проектування				

<p>використовують під час викладання курсу</p>	<p>алгоритмів і програм, виконання практичних завдань та проєктів; 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.</p>
<p>Необхідне обладнання</p>	<p>Для проведення лекцій: комп'ютер, проектор, доступ до мережі інтернет. Для проведення лабораторних та виконання завдань: комп'ютер; ОС Windows/Linux; доступ до інтернету; середовища програмування мовою C++ (Microsoft Visual Studio, Code Blocks тощо). 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.</p>
<p>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</p>	<p>Оцінювання 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). На екзамен студент отримує одну комплексну задачу, для якої треба програмно реалізувати низку вимог за темами прочитаного курсу. Перелік вимог, шкала оцінювання і зразок завдання будуть надані студентам до закінчення лекційних і лабораторних занять. Оцінка за екзаменаційне завдання може отримати додаткові бали, якщо завдання чи проєкти студента, виконані впродовж семестру, мали закінчений характер і високий рівень фахової реалізації. Відвідання занять 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. Академічна доброчесність: 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>
<p>Опитування</p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.</p>