

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Фізичний факультет
Кафедра фізики твердого тіла

Затверджено

На засіданні кафедри фізики твердого тіла
фізичного факультету
Львівського національного університету
імені Івана Франка
(протокол № 1 від 25 серпня 2023 р.)

Завідувач кафедри



проф. Капустяник В.Б..

Силабус

**з навчальної дисципліни «Об'єктно-орієнтоване програмування (ООП)»,
що викладається в межах
ОПП «Комп'ютерна фізика»
першого (бакалаврського) рівня вищої освіти
для здобувачів з спеціальності 104 Фізика та астрономія**

Львів 2023 р.

**Силабус курсу «ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»
2023–2024 н.р.**

Назва курсу	Об'єктно-орієнтоване програмування
Адреса викладання курсу	вул. Кирила і Мефодія 8, 79005 Львів
Факультет та кафедра, за якою закріплена дисципліна	фізичний факультет, кафедра фізики твердого тіла
Галузь знань, шифр та назва спеціальності	10 Природничі науки / 104 Фізика та астрономія
Викладачі курсу	доцент кафедри фізики твердого тіла, к.ф.-м.н Бовгира Олег Вікторович
Контактна інформація викладачів	oleh.bovhyra@lnu.edu.ua , yuriy.eliyashevskyy@lnu.edu.ua https://physics.lnu.edu.ua/employee/bovhyra-oleh-viktorovych
Консультації по курсу відбуваються	Консультації в день проведення лекцій та лабораторних занять (за попередньою домовленістю). Також можливі он-лайн консультації через електронну пошту та на платформі Microsoft Teams. Для погодження часу он-лайн консультацій слід написати на електронну пошту викладача або в чат Microsoft Teams
Сторінка курсу	https://physics.lnu.edu.ua/course/ooop-prykladna-fizyka
Інформація про курс	Дисципліна «Об'єктно-орієнтоване програмування» є нормативною дисципліною з спеціальності 104 Фізика та астрономія (спеціалізація “Комп'ютерна фізика”) для першого (бакалаврського) рівня вищої освіти, яка викладається в 3 семестрі в обсязі 5 кредитів (за Європейською Кредитно-Трансферною Системою ECTS). Курс розроблено таким чином, щоб надати учасникам відповідні теоретичні знання, уміння, навички, загальні та фахові компетентності для продукування нових ідей, розв'язання комплексних проблем у галузі обчислювальної фізики.
Коротка анотація курсу	Об'єктно-орієнтоване програмування є одним із основних підходів до розробки великих програмних проектів при розв'язку прикладних та наукових задач. Розуміння його концепції, переваг та недоліків в комплексі із практичними навичками розробки та написання програм в рамках цієї парадигми є важливою складовою підготовки сучасного фахівця. У результаті вивчення дисципліни в студентів формуються навички проектування, програмування й налагодження об'єктно-орієнтованих програм мовою програмування C++ та C#. Зміст курсу: <ul style="list-style-type: none"> • Основи об'єктно-орієнтованої мови програмування • Поняття об'єктно-орієнтованого аналізу, проектування та програмування • Повторне використання класів • Принципи об'єктно-орієнтованого проектування класів • Оброблення виняткових ситуацій • Стандартні бібліотеки класів середовищ розробника програм

<p>Мета та цілі курсу</p>	<p>Метою і завданням навчальної дисципліни є засвоєння необхідних знань із основ об'єктно - орієнтованого програмування, отримання навичок використання класів, механізмів наслідування, інкапсуляції та поліморфізму; отримання практичних навичок самостійно будувати програми середнього рівня складності з використанням об'єктно – орієнтованої парадигми програмування; застосування здобутих знань та підходів для розв'язання практичних задач різного рівня складності, що в подальшому стане цінним інструментом у майбутній професійній діяльності.</p>
<p>Література для вивчення дисципліни</p>	<p>Базова:</p> <ol style="list-style-type: none"> 1. Бублик В.В. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТкнига, 2015. – 624 с. 2. Кравець П.О. Об'єктно-орієнтоване програмування: навч. посібник / П.О. Кравець. – Львів: Видавництво Львівської політехніки, 2012. – 624 с. 3. Жуковський С.С. Об'єктно-орієнтоване програмування мовою С++. / Жуковський С.С., Вакалюк Т.А. Навчально-методичний посібник. – Житомир: Вид-во ЖДУ, 2016. – 100 с. 4. Коноваленко І.В. Програмування мовою С# 6.0 / Коноваленко І.В. – Тернопіль, ТНТУ, 2016 – 227 ст. 5. Васильєв О. Програмування на С++ в прикладах і задачах : Навч. посіб. / О. Васильєв. – Київ : Видавництво Ліра-К, 2017. – 382 с. <p>Допоміжна:</p> <ol style="list-style-type: none"> 1. Грицюк Ю.І. Об'єктно-орієнтоване програмування мовою С++: навчальний посібник / Ю.І. Грицюк, Т.Є. Рак. – Львів: Львівського ДУ БЖД, 2011. – 404 с. 2. Бойко Б. І. Об'єктно-орієнтоване програмування. Лабораторний практикум: навчальний посібник / Б. І. Бойко, Л. Л. Омельчук, Н. Г. Русіна – К.: «Айс Принт», 2016. – 90 с. 3. Stroustrup B. The Design and Evolution of C / Дизайн і еволюція С++. 4. Meyers S. Effective STL / Ефективне використання STL. 5. Настенко, Д. В. Об'єктно-орієнтоване програмування. Частина 1. Основи об'єктно-орієнтованого програмування на мові С# [Електронний ресурс] : навчальний посібник / Д. В. Настенко, А. Б. Нестерко ; НТУУ «КПІ». – Електронні текстові дані (1 файл: 931,2 Кбайт). – Київ : НТУУ «КПІ», 2016. – 76 с. 6. Мартін, Роберт. Чиста архітектура. Мистецтво розроблення програмного забезпечення [Текст] / Р. Мартін ; пер. з англ. І. Бондаря-Терещенка. - Харків : Фабула : Ранок, 2020. - 368 с. 7. Фрімен Ерік. Head First. Патерни проектування / Ерік Фрімен, Елізабет Робсон, Кеті Сьєрра і Берт Бейтс; пер. з англ. Г. Якубовська — Харків : ВД «Фабула», 2020. – 672 с. <p>Інформаційні ресурси:</p> <ol style="list-style-type: none"> 1. Системи онлайн-освіти: <ol style="list-style-type: none"> a. https://prometheus.org.ua/, b. https://www.coursera.org, http://www.udacity.com, 2. https://www.bestprog.net/uk/sitemap_ua/c/ 3. https://www.tutorialspoint.com/cplusplus/cpp_object_oriented.htm

	<p>4. https://www.learncpp.com/cpp-tutorial/</p> <p>5. https://www.codesdope.com/cpp-oop/</p> <p>6. https://elearning.sumdu.edu.ua/free_content/lectured:035d976fe705881ad71ba486aa8d45ac0b3ce889/20200921190923//index.html</p>
Тривалість курсу	один семестр
Обсяг курсу	5 кредитів ЄКТС, 150 годин, з яких 80 годин аудиторних занять, з них 32 годин лекцій, 48 годин лабораторних занять, та 70 годин самостійної роботи
Очікувані результати навчання	<p>В результаті вивчення цього курсу здобувач буде знати: фундаментальні концепції ООП; принципи програмування та реалізації ООП. мови об'єктно-орієнтованого програмування C++; переважання операторів; наслідування; поліморфізм; шаблони; методи обробки виняткових ситуацій; ієрархію класів; організацію стандартної бібліотеки і контейнерів.</p> <p>вміти: визначати класи та об'єкти при застосуванні об'єктно-орієнтованих технологій програмування; здійснювати об'єктно-орієнтовану декомпозицію предметної області в умовах програмування складних об'єктів і систем за допомогою процедур об'єктно-орієнтованого аналізу, об'єктно-орієнтованого програмування, використовуючи визначення класів, поведінки об'єктів, структури даних та їх взаємозв'язки; розробляти об'єктно-орієнтовану модель предметної галузі за допомогою мов об'єктно-орієнтованого моделювання в умовах програмування складних об'єктів та систем, використовуючи об'єктно-орієнтовану нотацію складних систем.</p> <p>В результаті вивчення цього курсу здобувач має оволодіти такими компетентностями:</p> <p>Загальні компетентності (ЗК):</p> <p>K01. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>K02. Здатність застосовувати знання у практичних ситуаціях.</p> <p>K03. Навички використання інформаційних і комунікаційних технологій.</p> <p>K12. Здатність спілкуватися державною мовою як усно, так і письмово.</p> <p>Спеціальні (фахові) компетентності (СК):</p> <p>K24. Здатність працювати з джерелами навчальної та наукової інформації.</p> <p>K25. Здатність самостійно навчатися і опановувати нові знання з фізики, астрономії та суміжних галузей.</p> <p>Після засвоєння навчальної дисципліни студенти мають продемонструвати такі програмні результати навчання:</p> <p>K30. Здатність планувати та створювати програмне забезпечення використовуючи мови високого рівня.</p> <p>K32. Здатність до планування розробки великих програмних продуктів шляхом поділу їх на окремі задачі.</p>
Ключові слова	інкапсуляція, наслідування, поліморфізм, віртуальні функції, обробка виняткових ситуацій
Формат курсу	Очний
	проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем

Теми	Наведено у табл.1 і табл. 2
Підсумковий контроль, форма	Підсумковий контроль: іспит в кінці семестру. Форма: усна.
Пререквізити	Для вивчення курсу необхідні знання з таких предметів: “Вища математика”, “Обчислювальна техніка і програмування”.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Використовуються такі методи навчання: а) <i>словесні</i> – лекція, пояснення, бесіда, інструктаж (вступний та поточний) під час виконання лабораторних робіт; б) <i>наочні</i> – ілюстрування лекційного матеріалу таблицями, схемами та графіками; в) <i>практичні</i> – виконання лабораторних робіт, що передбачає організацію навчальної роботи для отримання нових знань.
Необхідне обладнання	персональний комп'ютер, операційні системи (Windows, Linux), спеціальне програмне забезпечення (Code::Blocks, Microsoft Visual Studio), загальнонавчальні комп'ютерні програми, проектор
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням: • лабораторні: 40% семестрової оцінки; максимальна кількість балів 40 • контрольні заміри (модулі): 10% семестрової оцінки; максимальна кількість балів 10 • іспит: 50% семестрової оцінки. Максимальна кількість балів 50 Підсумкова максимальна кількість балів 100. Академічна доброчесність: очікується, що роботи студентів будуть їхніми оригінальними дослідженнями чи міркуваннями. Списування, втручання в роботу інших студентів, відсутність посилань на використані джерела становлять, але не обмежують, приклади можливої академічної не доброчесності. Виявлення ознак академічної не доброчесності в роботах студента є підставою для її не зарахування викладачем, незалежно від масштабів плагіату чи обману. Відвідування занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти мають інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів робіт, передбачених курсом. Література: уся література, яку студенти не зможуть знайти самостійно. Буде надана викладачем виключно в освітніх цілях без права передачі її третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих. Політика встановлення балів. Враховуються бали набрані за виконання та захист лабораторних робіт і самостійної роботи. При цьому враховується присутність на заняттях та активність студента під час виконання лабораторної роботи; списування та плагіат; користування мобільними пристроями в цілях не пов'язаних з навчанням; несвоєчасне виконання поставленого завдання. Жодні форми порушення академічної доброчесності не толеруються.
Питання до екзамену	1. Парадигми програмування. Особливості об'єктно-орієнтованої парадигми. Мови програмування, що її підтримують. 2. Поняття класу та об'єкту. Відповідні конструкції C++. 3. Поля та методи класу. Відповідні конструкції C++.

	<ol style="list-style-type: none"> 4. Конструктори та деструктори: призначення, види, особливості. 5. Створення та знищення об'єктів в C++. Стек та динамічна пам'ять. 6. Інкапсуляція. Її реалізація у C++. Специфікатори доступу. 7. Поняття області видимості. Простори імен. 8. Вкладені класи. Особливості визначення та використання. 9. Специфікатори доступу в C++. Дружні класи та функції. 10. Константні та статичні члени класу. 11. Перевантаження методів. Поняття сигнатури методу. 12. Перевантаження операторів. Реалізація та існуючі обмеження в C++. 13. Агрегація та композиція. Об'єкти, посилання та вказівники на них як поля класів C++. 14. Вказівники та "розумні" вказівники (на прикладі auto_ptr). 15. Композиція в C++. Створення, ініціалізація та знищення вкладених об'єктів. 16. Наслідування, його типи. Специфікатори доступу при наслідуванні. 17. Множинне наслідування. Особливості та проблеми використання. 18. Поліморфізм. Способи його реалізації в C++. 19. Віртуальні методи, абстрактні класи. 20. Статичне та динамічне зв'язування. 21. Приведення типів у C++. 24. Підтримка узагальненого програмування в C++. Шаблони. 25. Шаблони класів, функцій, методів. 26. Нетипові параметри шаблонів. 27. Склад та можливості стандартної бібліотеки шаблонів STL. 28. Послідовні контейнери STL. Їх відмінності від масивів у стилі C та між собою. 29. Асоціативні контейнери STL. Їх відмінності. 30. Ітератори STL. Призначення та види. 31. Основи об'єктно-орієнтованого проектування програмного забезпечення мовою UML. 32. Використання UML та шаблонів у процесі об'єктно-орієнтованого аналізу та проектування.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

Схема курсу «Об'єктно-орієнтоване програмування»

Тиждень	Назва теми	Форма діяльності та обсяг годин	Література	Термін виконання
1	Вступ Структурне, модульне, подійно-орієнтоване та об'єктно-орієнтоване програмування. Порівняльна характеристика. Історія об'єктно-орієнтованого програмування. Основні поняття. Абстракція даних. Наслідування, інкапсуляція та поліморфізм.	Лекції – 2 год., самос. робота – 1 год.	Базова: 1, 2, 3; Допоміжна: 1, 3	1 тиждень
2	Тема 1. Структури Структури. Масиви структур. Вказівники на структури. Динамічні структури даних. Лінійні списки.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 1, 3	1 тиждень
3	Тема 2. Об'єкти та класи Поняття об'єкта в ООП. Стан та поведінка об'єкта. Відношення між об'єктами. Поняття класу. Типи класів. Відношення між класами.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 1, 3, 5	1 тиждень
4	Тема 3. Створення і знищення об'єктів Поняття класу та його визначення. Специфікатори public, private. Доступ до відкритих членів класу. Функції доступу до захованих членів класу. Створення об'єктів. Статичні члени класу. Дружні і складові функції. Дружні класи. Ініціалізація та знищення об'єктів. Автоматичне і динамічне виділення пам'яті під об'єкти класу. Конструктори і деструктори. Конструктор копіювання.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 1, 3, 5	1 тиждень
5	Тема 4. Успадкування Похідні класи. Конструктори похідних класів. Виклик конструкторів похідних класів. Виклик конструкторів ієрархічних класів. Множинне наслідування. Виклик конструкторів при множинному успадкуванні. Передача параметрів конструктору базового класу. Перетворення вказівників на об'єкти класів. Приклади успадкування.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 1, 3, 5	1 тиждень
6	Тема 5. Поліморфізм. Віртуальні функції Віртуальні функції та поліморфізм. Чисто віртуальні функції та абстрактні класи. Механізм пізнього зв'язування. Створення бібліотеки класів графічних фігур.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 1, 3, 5	1 тиждень
7	Тема 6. Перевантаження операцій Перевантаження операцій для класів. Бінарні і унарні операції. Пріоритети при перевантаженні операцій. Присвоєння і ініціалізації. Перезавантаження операторів виклику функції, індексування. Стандартні об'єкти потокового вводу-виводу C++.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 1, 3	1 тиждень

	Перезавантаження операцій “<<”, “>>” для класу користувача. Приклади проектування і побудови класів.			
8	Тема 7. Шаблонні функції і класи Узагальнення у функціях. Поняття шаблонної функції. Поняття шаблону класу. Ключове слово <code>template</code> . Переваги використання шаблонів.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 3, 4, 6	1 тиждень
9	Тема 8. Стандартна бібліотека шаблонів STL Склад STL. Предикати та функції-об’єкти. Алокатори. Ітератори. Поняття ітератора.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 4	1 тиждень
10	Тема 9. Контейнерні класи Поняття контейнера. Типи контейнерів. Контейнер <code>vector</code> . Контейнер <code>deque</code> . Асоціативні контейнери. Контейнер <code>Set</code> . Контейнер <code>Map</code> .	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 3, 4, 6	1 тиждень
11	Тема 10. Стандартна бібліотека <code>algorithm</code> Модифікуючі та немодифікуючі операції. Алгоритми сортування. Алгоритми роботи з множинами.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 4, 7	1 тиждень
12-13	Тема 11. Обробка виключних ситуацій. Обробка помилок. Розпізнавання ситуацій. Класифікація ситуацій. Використання ресурсів. Генерація виняткових ситуації. Повторне генерування виняткової ситуації. Функції <code>terminate()</code> і <code>unexpected()</code> . Обробники, пов’язані з функціями <code>terminate()</code> і <code>unexpected()</code> . Класи <code>exception</code> і <code>bad_exception</code> . перехоплення класів виключних ситуацій. Застосування декількох операторів <code>catch</code> . Обробка похідних виключних ситуацій.	Лекції – 4 год., самос. робота – 3 год.	Базова: 2, 3; Допоміжна: 3, 4	2 тижні
14	Тема 12. Об’єктно-орієнтований аналіз Об’єктно-орієнтований аналіз і декомпозиція складних об’єктів. Об’єктно-орієнтований синтез інформаційних моделей. Використання мови об’єктно-орієнтованого моделювання UML.	Лекції – 2 год., самос. робота – 2 год.	Базова: 2, 3; Допоміжна: 3, 7	1 тиждень
15	Тема 13. Платформа .NET та мова C# Історія та причини виникнення мови програмування C#. Особливості реалізації об’єктно-орієнтованого підходу у мові програмування C#. Приклади програмного коду на мові C#. C# .Net. Розширені можливості реалізації ООП у мові C#. Події.	Лекції – 2 год., самос. робота – 2 год.	Базова: 4; Допоміжна: 5	1 тиждень
16	Тема 14. Графічний інтерфейс в C# Створення графічного інтерфейсу користувача та програмування для Windows Forms у середовищі C#. Елементи керування, їх властивості, методи, події. Механізм Drag-and-Drop. Векторна та растрова графіка. Графічна підсистема C#. Використання XML, SVG та мови ECMAScript для Web-інтерфейсів.	Лекції – 2 год., самостійна робота – 2 год.	Базова: 4; Допоміжна: 5	1 тиждень

Теми лабораторних занять

Тиждень	Назва теми	Форма діяльності та обсяг годин	Термін виконання
1	Структури. Зовнішні функції і функції-члени структури.	лабор. заняття – 3 год., самостійна робота – 3 год.	1 тиждень
2	Робота з пам'яттю. Статична і динамічна пам'ять.	лабор. заняття – 3 год., самостійна робота – 3 год.	1 тиждень
3	Розробка програм з використанням найпростіших класів та об'єктів.	лабор. заняття – 3 год., самостійна робота – 3 год.	1 тиждень
4	Створення класів. Конструктори, деструктори, дружні функції і члени класу.	лабор. заняття – 3 год., самостійна робота – 3 год.	1 тиждень
5-6	Успадкування в C++.	лабор. заняття – 6 год., самостійна робота – 4 год.	2 тижні
7	Поліморфізм.	лабор. заняття – 3 год., самостійна робота – 3 год.	1 тиждень
8	Перевантаження операторів.	лабор. заняття – 3 год., самостійна робота – 3 год.	1 тиждень
9	Розробка шаблонів.	лабор. заняття – 3 год., самостійна робота – 3 год.	1 тиждень
10	Обробка виключних ситуацій.	лабор. заняття – 3 год., самостійна робота – 3 год.	1 тиждень
11-12	Використання бібліотеки STL.	лабор. заняття – 6 год., самостійна робота – 4 год.	2 тижні
13-14	Об'єктно-орієнтований аналіз і декомпозиція складних об'єктів з використанням мови UML.	лабор. заняття – 6 год., самостійна робота – 4 год.	2 тижні
15-16	Windows Forms. Елементи форми та їхні властивості.	лабор. заняття – 6 год., самостійна робота – 4 год.	2 тижні