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. Володимир КАПУСТЯНИК

Силабус

з навчальної дисципліни «Об'єктно-орієнтоване програмування (ООП)»,
що викладається в межах
ОПП «Комп'ютерні технології в прикладній фізиці»
першого (бакалаврського) рівня вищої освіти
для здобувачів з спеціальності 105 Прикладна фізика та наноматеріали

Львів 2023

Назва дисципліни	**Об’єктно-орієнтоване програмування**
Адреса викладання дисципліни	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	Галузь знань – 10 Природничі науки Спеціальність – 105 Прикладна фізика та наноматеріали
Викладачі дисципліни	доцент кафедри фізики твердого тіла, к.ф.-м.н Бовгира Олег Вікторович доцент кафедри фізики твердого тіла, к.ф.-м.н Еліяшевський Юрій Ігорович
Контактна інформація викладачів	oleh.bovhyra@lnu.edu.ua, yuriy.eliyashevskyy@lnu.edu.ua https://physics.lnu.edu.ua/employee/bovhyra-oleh-viktorovych
Консультації по дисципліні 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 дисципліни	https://physics.lnu.edu.ua/course/oop-prykladna-fizyka
Інформація про дисципліну	Дисципліна «Об’єктно-орієнтоване програмування» є нормативною дисципліною з спеціальності 105 Прикладна фізика та наноматеріали (ОПП “Комп’ютерні технології в прикладній фізиці”) для першого (бакалаврського) рівня вищої освіти, яка викладається в 3 семестрі в обсязі 3 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Об’єктно-орієнтоване програмування є одним із основних підходів до розробки великих програмних проектів під час 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#. 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. **Зміст курсу**: • 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
Мета та цілі дисципліни	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.
Література для вивчення дисципліни	**Базова:** 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. Ярошко С.А. Методи розробки алгоритмів. Програмування мовою С++ : навчальний посібник / С.А. Ярошко, О.С. Ярошко – Львів : ЛНУ імені Івана Франка, 2022. – 248 с. 5. Алхімова С. М. Обєктно-орієнтоване програмування : підручник. У 2-х ч. Ч. 2. Об'єктно-орієнтований підхід до розробки програмного забезпечення / С. М. Алхімова. – Київ: КГП ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. – 192 с. 6. Васильєв О. Програмування на С++ в прикладах і задачах : Навч. посіб. / О. Васильєв. – Київ : Видавництво Ліра-К, 2017. – 382 с. **Допоміжна:** 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. Мартін, Роберт. Чиста архітектура. Мистецтво розроблення програмного забезпечення [Текст] / Р. Мартін ; пер. з англ. І. Бондаря-Терещенка. - Харків : Фабула : Ранок, 2020. - 368 с. 6. Фрімен Ерік. Head First. Патерни проєктування / Ерік Фрімен, Елізабет Робсон, Кеті Сьєрра і Берт Бейтс; пер. з англ. Г. Якубовська – Харків : ВД «Фабула», 2020. – 672 с. **Інформаційні ресурси:** 1. Системи онлайн-освіти: 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 4. https://www.learncpp.com/cpp-tutorial/ 5. https://www.codesdope.com/cpp-oop/
Тривалість курсу	один семестр
Обсяг курсу	3 кредити, 90 годин, з яких 48 годин аудиторних занять, з них 16 годин лекцій, 32 годин практичних занять, та 42 години самостійної роботи
Очікувані результати навчання	Курс формує такі **загальні** (ЗК) та **спеціальні компетентності** (СК): ЗК 5. Навички використання інформаційних і комунікаційних технологій. ЗК 6. Здатність до проведення досліджень на відповідному рівні.

	ЗК 7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел. СК 5. Здатність до постійного розвитку компетентностей у сфері прикладної фізики, інженерії та комп'ютерних технологій. Після засвоєння навчальної дисципліни студенти мають продемонструвати такі **програмні результати навчання**: ПРН 03. Застосовувати ефективні технології, інструменти та методи експериментального дослідження властивостей речовин і матеріалів, включаючи наноматеріали, при розв'язанні практичних проблем прикладної фізики. ПРН 07. Класифікувати, аналізувати та інтерпретувати науково технічну інформацію в галузі прикладної фізики. ПРН 9. Презентувати результати досліджень і розробок фахівцям і нефахівцям, аргументувати власну позицію. ПРН 11. Знати цілі сталого розвитку та можливості своєї професійної сфери для їх досягнення, в тому числі в Україні. ПРН 14. Програмувати з використанням мов високого рівня. В результаті вивчення цього курсу здобувач буде **знати:** фундаментальні концепції ООП; принципи програмування та реалізації ООП. мови об'єктно-орієнтованого програмування С++; перевантаження операторів; наслідування; поліформізм; шаблони; методи обробки виняткових ситуацій; ієрархію класів; організацію стандартної бібліотеки і контейнерів. **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.
Ключові слова	інкапсуляція, наслідування, поліморфізм, віртуальні функції, обробка виняткових ситуацій
Формат курсу	Очний
	проведення лекцій, практичних занять 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, письмова.
Пререквізити	Для вивчення курсу необхідні знання з таких предметів: "Аналітична геометрія", "Обчислювальна техніка і програмування".
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Використовуються такі методи навчання: а) *словесні* – лекція, пояснення, бесіда, інструктаж (вступний та поточний) під час виконання практичних робіт; б) *наочні* – ілюстрування лекційного матеріалу слайдами, блок-схемами та кодами програм; в) *практичні* – виконання практичних робіт, що передбачає організацію навчальної роботи із використанням обчислювальної техніки.
Необхідне обладнання	персональний комп'ютер, операційні системи (Windows, Linux), спеціальне програмне забезпечення (Code::Blocks, Microsoft Visual

	Studio), загальновживані комп'ютерні програми, проектор
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням: • практичні: 12 практичних робіт, 40% семестрової оцінки; максимальна кількість балів 40 (практ. роботи № 1-4, 6-9 по 3 бали, № 5, 10-12 по 4 бали); • контрольні заміри (тестові модулі): 10% семестрової оцінки; максимальна кількість балів 10; • іспит: 50% семестрової оцінки. Максимальна кількість балів 50 (теоретична частина (2×5 = 10 балів), тест (20 балів) і написання програм (2×10 = 20 балів)). Підсумкова максимальна кількість балів 100. **Критерії оцінювання практичних робіт:** 90-100 % б. – студент повністю виконав завдання і володіє матеріалом на високому рівні; 60-80 % б. – студент повністю виконав завдання і володіє матеріалом на достатньому рівні; 30-50 % б. – студент частково виконав завдання і володіє матеріалом на задовільному рівні; 10-20 % б. – студент частково виконав завдання і тільки частково володіє матеріалом; 0 б. – студент не виконав завдання. **Критерії оцінювання описових питань іспиту:** 5 б. – студент повністю володіє матеріалом; 4 б. – студент достатньо володіє матеріалом; 3 б. – студент частково володіє матеріалом; 1-2 б. – студент має базові знання матеріалу; 0 б. – студент не володіє матеріалом. **Академічна доброчесність** здобувачами вищої освіти передбачає самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання. Списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій чи усній роботі студента є підставою для її незарахування викладачем, незалежно від масштабів обману. **Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і практичні 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**. Усю література, яку студенти не зможуть знайти самостійно, буде надано викладачами виключно в освітніх цілях без права її передавання третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих. **Політика виставлення балів**. Враховуються бали набрані за виконання та захист практичних робіт і самостійної роботи. При цьому враховується присутність на заняттях та активність студента під час виконання практичної 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 С++.

	3. Поля та методи класу. Відповідні конструкції C++. 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.

Таблиця 1

Схема курсу «**Об’єктно-орієнтоване програмування**»

Тиж-день	Назва теми	Форма діяльності та обсяг годин	Література	Термін виконання
1-2	**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. **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, 4; Допоміжна: 1, 3	2 тижні

3-4	**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. **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: 1-6; Допоміжна: 1, 3	2 тижні
5-6	**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: 1-6; Допоміжна: 1, 3	2 тижні
7-8	**Тема 5. Поліморфізм. Віртуальні функції** Віртуальні функції та поліформізм. Чисто віртуальні функції та абстрактні класи. Механізм пізнього зв'язування. Створення бібліотеки класів графічних фігур. **Тема 6. Перевантаження операцій** Перевантаження операцій для класів. Бінарні і унарні операції. Пріоритети при перевантаженні операцій. Приклади проектування і побудови класів.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1-6; Допоміжна: 1, 3	2 тижні
9-10	**Тема 7. Шаблонні функції і класи** Узагальнення у функціях. Поняття шаблонної функції. Поняття шаблону класу. Ключове слово template. Переваги використання шаблонів. **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: 1-5; Допоміжна: 1, 3, 4, 5, 6	2 тижні
11-12	**Тема 9. Контейнерні класи** Поняття контейнера. Типи контейнерів. Контейнер vector. Контейнер deque. Асоціативні контейнери. Контейнер Set. Контейнер Map. **Тема 10. Стандартна бібліотека algorithm** Модифікуючі та немодифікуючі операції. Алгоритми сортування. Алгоритми роботи з множинами.	Лекції – 2 год., самос. робота – 2 год.	Базова: 1, 2, 3; Допоміжна: 1, 3, 5, 6	2 тижні
13-14	**Тема 11. Обробка виключних ситуацій.** Обробка помилок. Розпізнавання ситуацій. Класифікація ситуацій. Використання ресурсів. Генерація виняткових ситуації. Повторне	Лекції – 2 год., самос. робота – 2 год.	Базова: 1-5; Допоміжна: 1, 3	2 тижні

	генерування виняткової ситуації. Функції terminate() і unexpected(). Обробники, пов'язані з функціями terminate() і unexpected(). Класи exception і bad_exception. Перехоплення класів виключних ситуацій. Застосування декількох операторів catch. Обробка похідних виключних ситуацій.			
15-16	**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: 1, 2, 3, 5; Допоміжна: 1, 3, 5, 6	2 тижні

Таблиця 2

Теми практичних занять

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