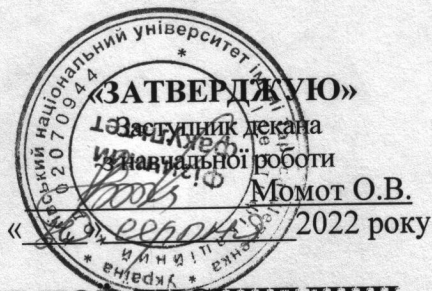


КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Фізичний факультет  
(назва факультету, інституту)

Кафедра ядерної фізики та високих енергій



**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**  
**Сучасні мови та об'єктно-орієнтоване програмування в ядерній фізиці**  
(повна назва навчальної дисципліни)

для студентів

галузь знань

**10 Природничі науки**  
(шифр і назва)

спеціальність

**104 – “Фізика та астрономія”**  
(шифр і назва спеціальності)

освітній рівень

**магістр**

(молодший бакалавр, бакалавр, магістр)

освітня програма

**Фізика високих енергій**  
(назва освітньої програми)

вид дисципліни

обов'язкова

Форма навчання

денна

Навчальний рік

2022/2023

Семестр

2

Кількість кредитів ECTS

6

Мова викладання, навчання  
та оцінювання

українська

Форма заключного контролю

іспит

Викладачі: канд. фіз.-мат. доцент КЯФ Р.В. Єрмоленко.

(Науково-педагогічні працівники, які забезпечують викладання даної дисципліни у відповідному навчальному році)

Пролонговано: на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» \_\_\_\_ 20\_\_ р.  
(підпис, ПІБ, дата)

на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» \_\_\_\_ 20\_\_ р.  
(підпис, ПІБ, дата)

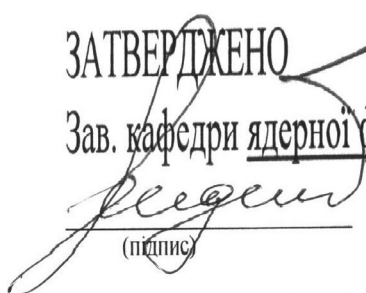
на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» \_\_\_\_ 20\_\_ р.  
(підпис, ПІБ, дата)

КИЇВ – 2022

Розробники: Р.В.Єрмоленко, канд. фіз.-мат. наук, доцент КЯФ

ЗАТВЕРДЖЕНО

Зав. кафедри ядерної фізики та високих енергій

 (підпис) ( Каденко І.М. )  
(прізвище та ініціали)

Протокол №14 від «03» червня 2022 р.

Схвалено науково - методичною комісією фізичного факультету

Протокол від « 10 » червня 2022 року № 11

Голова науково-методичної комісії

 ( Оліх. О.Я. )  
(прізвище та ініціали)

## **ВСТУП**

**1. Мета дисципліни** – Метою дисципліни «Сучасні мови та об'єктно-орієнтоване програмування в ядерній фізиці» є надання студентам:

- необхідних теоретичних відомостей з об'єктно орієнтованої парадигми програмування;
- практичних навичок розробки програмного забезпечення з використанням C++;
- Знання бібліотеки STL, стандартів C++ 11, C++ 14.
- Алгоритмів з застосуванням C++.

**2. Попередні вимоги до опанування або вибору навчальної дисципліни:**

- Успішне опанування основних курсів фізики: «Математичний аналіз», «Лінійна алгебра», «Програмування», «Обчислювальні методи».
- Вміти створювати імперативні програми.
- Володіти навичками роботи на комп'ютері по пошуку інформації в мережі Інтернет.

**3. Анотація навчальної дисципліни:**

Навчальна дисципліна "Сучасні мови та об'єктно-орієнтоване програмування в ядерній фізиці" є складовою циклу професійної підготовки фахівців освітньо-кваліфікаційного рівня "магістр".

Курс «Сучасні мови та об'єктно-орієнтоване програмування в ядерній фізиці» дозволить значно покращити професійну підготовку студентів кафедри ядерної фізики, що пов'язано з тим, що студенти будуть:

- Знати сучасні стандарти C++.
- Розуміти Об'єктно орієнтовану парадигму програмування;
- Вміти реалізовувати базові алгоритми;
- Створювати програми з використанням сучасних підходів та методів ООП.

**4. Завдання (навчальні цілі)** – Спецкурс "Сучасні мови та об'єктно-орієнтоване програмування в ядерній фізиці" дозволить студентам оволодіти методологією ООП C++, знати бібліотеку STL, знати сучасні алгоритми. Згідно вимог Стандарту вищої освіти України (другий (магістерський) рівень вищої освіти, галузь знань 10 «Фізика», спеціальність 104 «Фізика та астрономія», ОНП «Фізика високих енергій» дисципліна забезпечує набуття здобувачами освіти наступних компетентностей:

Інтегральних:

Здатність розв'язувати складні задачі і проблеми дослідницького та інноваційного характеру у фізиці та астрономії.

Загальних:

**ЗК04.** Здатність вчитися і оволодівати сучасними знаннями.

**ЗК06.** Здатність виявляти, ставити та вирішувати проблеми.

**ЗК07.** Здатність проведення досліджень на відповідному рівні.

Фахових:

**СК03.** Здатність презентувати результати проведених досліджень, а також сучасні концепції фізики та астрономії фахівцям і нефахівцям.

**СК10.** Здатність використовувати знання й уміння в галузі практичного використання комп'ютерних технологій для дослідження процесів в ФВЕ.

**СК12.** Здатність застосовувати знання теорій опису фізичних властивостей елементарних частинок та процесів взаємодії.

## 5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація*; 4. автономність та відповідальність*)		Методи викладання і навчання	Методи оцінювання	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Знати принципи ООП	лекція	тест	50
2.1	Вміти створювати ПЗ з використанням C++ 11; Вміти реалізовувати алгоритми на C++	Практичні завдання	тест	50

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання (необов'язково для вибірових дисциплін)

Результати навчання дисципліни	1.1	2.1
<b>Програмні результати навчання</b>		
<b>РН04.</b> Вибирати та використовувати відповідні методи обробки та аналізу даних в фізичних та астрономічних дослідженнях і оцінювання їх достовірності.	+	
<b>РН12.</b> Розробляти та застосовувати ефективні алгоритми та спеціалізоване програмне забезпечення для дослідження моделей фізичних та астрономічних об'єктів і процесів, обробки результатів експериментів і спостережень		+
<b>РН13.</b> Створювати фізичні, математичні і комп'ютерні моделі природних об'єктів та явищ, перевіряти їх адекватність, досліджувати їх для отримання нових висновків та поглиблення розуміння природи, аналізувати обмеження.		+
<b>РН15.</b> Планувати наукові дослідження з урахуванням цілей та обмежень, обирати ефективні методи дослідження, робити обґрунтовані висновки за результатами дослідження.	+	+
<b>РН16.</b> Брати продуктивну участь у виконанні експериментальних та теоретичних досліджень в області фізики та астрономії.	+	+
<b>РН18.</b> Застосовувати сучасні методи програмування на мові C, C++ та Python з пакетом ROOT для розв'язування конкретних задач у фізиці високих енергій.	+	+
<b>РН23.</b> Вміти встановлювати причинно-наслідковий зв'язок між статичними та динамічними характеристиками частинок.	+	+

\*

\*

## 8. Схема формування оцінки:

**8.1 Форми оцінювання студентів:** (зазначається перелік видів робіт та форм їх контролю / оцінювання із зазначенням Min. – рубіжної та Max. кількості балів чи відсотків)

- семестрове оцінювання:

1. Опитування при проведенні лекційних занять (максимум – 50 балів).
2. Лабораторні роботи (максимум – 50 балів).

- Підсумкове оцінювання у формі іспиту (підсумкова кількість балів з дисципліни (максимум 100 балів), яка визначається як сума (проста або зважена) балів за систематичну роботу впродовж семестру.

	Семестрова кількість балів	ПКР (підсумкова контрольна робота) чи/або іспит	Підсумкова оцінка
Мінімум	40	20	60
Максимум	100	40	100

Студент не допускається до екзамену, якщо під час семестру набрав менше 20 балів.<sup>1</sup> для допуску до екзамену обов'язково. Оцінка за іспит не може бути меншою **20 балів** для отримання загальної позитивної оцінки за курс.

## 8.2 Організація оцінювання:

При оцінюванні семестрової кількості балів результати практичних занять мають ваговий коефіцієнт 0.7, поточний контроль шляхом опитування -0.3.

### Шкала відповідності

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно з можливістю повторного складання / Fail	35-59
Незадовільно з обов'язковим повторним вивченням дисципліни / Fail	0-34
Зараховано / Passed	60-100
Не зараховано / Fail	0-59

<sup>1</sup> У випадку, коли дисципліна завершується екзаменом не менше – **20** балів, а рекомендований мінімум **не менше 36 балів**, оскільки якщо студент на екзамені набрав менше **24 балів** (а це 60% від 40 балів, відведених на екзамен), то вони **не додаються** до семестрової оцінки незалежно від кількості балів, отриманих під час семестру, а в екзаменаційній відомості у графі «результуюча оцінка» переноситься лише кількість балів, отриманих під час семестру.

## СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ

№ п/п	Назва теми	Кількість годин		
		лекції	Практичні заняття	Самостійна робота
1	Вступ. Парадигми програмування. Об'єктно-орієнтоване програмування: підхід до організації програми.	3	3	12
2	Об'єкти та класи. Наслідування. Інкапсуляція. Поліморфізм.	3	3	12
3	Поняття про віртуальні функції та Поліморфізм	3	3	12
4	Організація класів і особливості роботи з Об'єктами	3	3	12
5	<i>Організація механізмів успадкування в Класах.</i>	3	3	12
6	<i>Робота з шаблонними функціями та класами .</i>	3	3	12
7	<i>Бібліотека STL</i>	3	3	12
8	<i>Сучасні стандарти C++</i>	3	3	12
9	<i>Алгоритми і їх реалізація.</i>	3	3	12
10	<i>Патерни при програмуванні</i>	3	3	12
	<b>ВСЬОГО</b>	<b>30</b>	<b>30</b>	<b>120</b>

**Загальний обсяг 180 год., в тому числі:**

Лекцій – 30 год.

Семінари – 0 год.

Практичні заняття – 30 год.

Лабораторні заняття - 0 год.

Тренінги - 0 год.

Консультації – 0 год.

Самостійна робота - 120 год.

### РЕКОМЕНДОВАНА ЛІТЕРАТУРА:

1. ISO/ISC DTR 19769 (April 5, 2011) FDIS, Standard for Programming Language C++.
2. ISO/IEC 14882:2014 — Information technology — Programming languages — C++.
3. Robert Sedgewick. Algorithms in C++.
4. Bjarne Stroustrup. Programming: Principles and Practice Using C++.
5. Stroustrup B. The Design and Evolution of C++.
6. Meyers S. Effective STL.
7. Alexandrescu A. Modern C++ Design: Generic Programming and Design Patterns Applied.
8. Steven C. McConnell. Code Complete: A Practical Handbook of Software Construction, Second Edition 2nd Edition.
9. A. Hunt, D. Thomas. The Pragmatic Programmer: From Journeyman to Master 1st Edition.
10. Thomas H. Cormen, Charles I. Lazerson, Ronald L. Rivest, Clifford Stein. Algorithms. Construction and Analysis.
11. Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. Algorithms.