

**КЛАСИЧНИЙ ПРИВАТНИЙ УНІВЕРСИТЕТ**  
**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ**

**СИЛАБУС**

**навчальної дисципліни**

**«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»**

**КОНТАКТНА ІНФОРМАЦІЯ ТА ТЕХНІЧНА ДОПОМОГА**  
(включаючи електронну пошту, робочий час / місцезнаходження тощо).

<b>Викладач (-і)</b>	Бречко Наталія Анатоліївна
<b>Контактний тел.</b>	+38(061) 228-07-85 (внутр. 224)
<b>E-mail:</b>	<a href="mailto:brechko.na@gmail.com">brechko.na@gmail.com</a>
<b>Сторінка курсу на сайті підтримки навчальних програм КПУ</b>	<a href="http://www.zhu.edu.ua/cpu_edu/course/view.php?id=2932">http://www.zhu.edu.ua/cpu_edu/course/view.php?id=2932</a>
<b>Консультації</b>	<i>Очні консультації:</i> за графіком консультацій викладача, а. 311, головний корпус КПУ <i>Консультації off-line:</i> шляхом повідомлення на сторінці навчальної дисципліни сайту підтримки навчальних програм КПУ <a href="http://www.zhu.edu.ua/cpu_edu/message">http://www.zhu.edu.ua/cpu_edu/message</a>

**АНОТАЦІЯ**

Навчальна дисципліна «Об'єктно-орієнтоване програмування» є нормативною для студентів другого (магістерського) рівня вищої освіти галузі знань 014 Середня освіта (освітня програма: Інформатика). Згідно з навчальним планом денної форми навчання вивчення дисципліни заплановано на 5 семестр (3 курс).

Мета дисципліни направлена на отримання студентами базових знань і практичних навиків основ сучасної технології розробки і реалізації складних і програмних продуктів. Отримані знання можуть бути використані в розробці системного і прикладного програмного забезпечення. Для засвоєння програми дисципліни необхідні знання, вміння та компетенції студентів, що мають початкову підготовку по інформатиці і програмуванню, знання основних базових алгоритмів і базових конструкцій мов програмування.

Освітній процес з дисципліни здійснюється за такими формами: навчальні заняття; самостійна робота; контрольні заходи. Видами навчальних занять згідно з навчальним планом є: лекції; практичні заняття, а також консультації.

Практичні заняття передбачають: вивчення призначення та способів використання автоматизації повсякденних завдань адміністрування, проблем та напрямків розвитку операційних систем, основних методів та засобів конфігурування й застосування різних сучасних операційних систем.

Самостійна робота студентів полягає у засвоєнні вивченого навчального матеріалу в час, вільний від обов'язкових навчальних занять, без участі викладача.

Консультації призначені для роз'яснення студентам теоретичних або практичних питань.

Засвоєння навчального матеріалу перевіряється за допомогою поточного контролю, який здійснюється на практичних заняттях у формі усних відповідей, самостійних робіт та підсумкових тематичних контрольних робіт.

Підсумковий (семестровий) контроль після завершення 5 семестру здійснюється у формі іспиту.

## ФОРМАТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**Загальна кількість годин – 120 год., у т. ч. 64 годин аудиторних занять і 56 годин самостійної роботи студента. Кількість кредитів ECTS – 4.**

Всього кредитів	Всього годин	Аудиторних годин	У тому числі			Сам. робота
			Лекц.	Лабор.	Семін. (практ.)	
4	120	64	16	48	-	56

## ОЗНАКИ ДИСЦИПЛІНИ

Курс (рік навчання)	Семестр	Цикл підготовки	Нормативна/вибіркова
3	5	професійна	нормативна

## МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### Мета навчальної дисципліни

Мета дисципліни направлена на отримання студентами базових знань і практичних навиків основ сучасної технології розробки і реалізації складних і програмних продуктів. Отримані знання можуть бути використані в розробці системного і прикладного програмного забезпечення. Для засвоєння програми дисципліни необхідні знання, вміння та компетенції студентів, що мають початкову підготовку по інформатиці і програмуванню, знання основних базових алгоритмів і базових конструкцій мов програмування.

### Завдання навчальної дисципліни

- основним завданням є вивчення та практичне впровадження базових принципів ООП, що забезпечують цілісність та гнучкість програмного коду:
- дисципліна спрямована на розвиток здатності структурувати складні задачі:
- завданням курсу є підготовка студента до розробки промислового коду, що відповідає критеріям масштабованості та підтримки:
- оволодіння інструментарієм сучасної розробки:

## ЗАПЛАНОВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

### У результаті вивчення дисципліни студенти повинні знати:

- ▲ основні ідеї і поняття об'єктно-орієнтованого підходу в розробці алгоритмів і програм, сучасні методи програмування;
- ▲ об'єктно-орієнтовані мови програмування C++, а так само практичне застосування програмування у візуальних середовищах програмування;
- ▲ основні, які широко використовуються, алгоритми обробки текстової інформації, основних обчислювальних методів і прийомів.

### Після вивчення дисципліни студенти повинні вміти:

- ▲ здійснювати розробку алгоритмів і програм з використанням об'єктно-орієнтованого підходу на мовах об'єктно-орієнтованого програмування;
- ▲ використовувати візуальні середовища програмування.

Відповідно до освітньо-професійної програми підготовки магістра галузі знань 014 Середня освіта (освітня програма: Інформатика).вивчення дисципліни «Об'єктно-орієнтоване програмування» сприяє формуванню компетентностей та програмних результатів навчання :

#### **Інтегральна компетентність:**

ІК Здатність розв'язувати складні спеціалізовані задачі у галузі середньої освіти, що передбачає застосування теоретичних знань і практичних умінь з наук предметної спеціальності, педагогіки, психології, теорії та методики навчання і характеризується комплексністю та невизначеністю умов організації освітнього процесу в закладах загальної середньої освіти.

#### **Загальні компетентності:**

ЗК 2. Здатність до абстрактного мислення, аналізу та синтезу. Володіння методами логічного та системного аналізу, здатність виявляти закономірності в інформаційних процесах та розв'язувати складні завдання предметної області.

ЗК 3. Інформаційно-комунікаційна компетентність. Здатність ефективно використовувати сучасні ІКТ для пошуку, обробки, критичної оцінки та поширення інформації в освітньому та соціальному просторах.

ЗК 7. Здатність до креативності та інновацій. Здатність генерувати нові ідеї, виявляти гнучкість у професійних ситуаціях та застосовувати творчий підхід до розв'язання практичних проблем.

#### **Спеціальні (фахові) компетентності**

СК 3. Програмно-технологічна компетентність. Здатність розробляти програмні продукти, використовуючи сучасні мови програмування, знання архітектури ПК, операційних систем та мереж.

#### **Програмні результати навчання:**

РН 3. Використовує сучасні мови, методи та технології програмування для розв'язання практичних завдань і створення програмних продуктів.

РН 7. Будує інформаційні та математичні моделі об'єктів, процесів і явищ; використовує бази даних та технології візуалізації для опрацювання і представлення даних.

## ПЛАН КУРСУ

Назва змістових модулів та тем	Лекц.	Практ (сем.)	Завдання для самостійної роботи
Тема № 1. “Основні особливості розробки об’єктно-орієнтованих програм мовою C++ .	2	8	1. Опрацювання лекційного матеріалу та питань самостійного вивчення. 2. Перевірка знань шляхом комп’ютерного тестування. 3. Підготовка звіту з виконання лабораторних робіт.
Тема №2. “Класи - основа об’єктно-орієнтованих програм”.	2	6	1. Опрацювання лекційного матеріалу та питань самостійного вивчення. 2. Перевірка знань шляхом комп’ютерного тестування. 3. Підготовка звіту з виконання лабораторних робіт.
Тема №3. "Організація класів і особливості роботи з об’єктами”.	4	10	1. Опрацювання лекційного матеріалу та питань самостійного вивчення. 2. Перевірка знань шляхом комп’ютерного тестування. 3. Підготовка звіту з виконання лабораторних робіт.
Тема №4. "Особливості механізму перевизначення операторів”	2	10	1. Опрацювання лекційного матеріалу та питань самостійного вивчення. 2. Перевірка знань шляхом комп’ютерного тестування. 3. Підготовка звіту з виконання лабораторних робіт.
Тема №5. "Організація механізмів успадкування у класах”.	4	6	1. Опрацювання лекційного матеріалу та питань самостійного вивчення. 2. Перевірка знань шляхом комп’ютерного тестування. 3. Підготовка звіту з виконання лабораторних робіт.
Тема №6. "Поняття про поліморфізм”	2	8	1. Опрацювання лекційного матеріалу та питань самостійного вивчення. 2. Перевірка знань шляхом комп’ютерного тестування. 3. Підготовка звіту з виконання лабораторних робіт.
Разом	16	48	56

### ФОРМИ КОНТРОЛЮ ТА КРИТЕРІЇ ОЦІНЮВАННЯ

У процесі вивчення навчальної дисципліни «Об’єктно-орієнтоване програмування» використовуються наступні види контролю:

1. Поточний контроль – здійснюється протягом семестру шляхом опитування на семінарських (практичних) заняттях, перевірки виконання тестових завдань, виконання лабораторних робіт, модульних контрольних робіт тощо. За змістом він включає перевірку ступеню засвоєння студентом навчального матеріалу, який охоплюється темою лекційного та семінарського заняття, уміння самостійно опрацьовувати навчально-методичну літературу, здатність осмислювати зміст теми, уміння публічно та письмово представити певний матеріал, а також виконання завдань самостійної роботи.

2. Підсумковий семестровий контроль – здійснюється у формі екзамену відповідно до графіка освітнього процесу.

Для оцінювання студентів використовується система накопичування балів. Згідно з «Положенням про організацію освітнього процесу в КПУ» підсумкова оцінка з дисципліни виставляється за 100-бальною шкалою з наступним переведенням у національну шкалу та шкалу ECTS. Бали нараховуються за виконання практичних завдань, лабораторних робіт, контрольних (модульних) завдань, тестів.

Результати поточного модульного контролю здобувачів вищої освіти є складовими елементами підсумкової оцінки з дисципліни.

Оцінка рівня роботи студента протягом семестру під час навчальних занять та самостійної роботи здійснюється у межах 60 балів. Вага екзамену у підсумковій оцінці складає 40 балів.

## РОЗПОДІЛ БАЛІВ ПОТОЧНОГО ТА ПІДСУМКОВОГО (СЕМЕСТРОВОГО) ОЦІНЮВАННЯ

Поточне оцінювання під час навчальних занять та самостійної роботи						Екзамен	Сума
Тема 1	Тема 2	Тема 3	Тема 4	Тема 5	Тема 6	40	100
10	10	10	10	10	10		

### Шкала оцінювання: 100-бальна, національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
75-81	C		
67-74	D	задовільно	
60-66	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

### РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Страуструп Б. Програмування: принципи та практика з використанням C++. — 2-ге вид. — К. : Діалектика, 2020. — 1328 с. (Базовий підручник від автора мови).
2. Прата С. Мова програмування C++. Лекції та вправи. — 6-ге вид. — К. : ТОВ "ІД Вільямс", 2017. — 1248 с.
3. Мейерс С. Ефективне використання C++. 55 порад щодо покращення структури та коду ваших програм. — К. : Діалектика, 2014. — 304 с. (Ключове видання для опанування нюансів ООП у C++).
4. Лафоре Р. Об'єктно-орієнтоване програмування в C++. — 4-ге вид. — К. 2013. — 928 с.
5. Гамма Е., Хелм Р., Джонсон Р., Вліссідес Дж. Прийоми об'єктно-орієнтованого проектування. Патерни проектування. — К. :, 2021. — 368 с. (Класика архітектурного мислення).
6. Мартин Р. Чистий код. Посібник із розробки програмного забезпечення. — К. : Фабула, 2019. — 368 с.

### ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

1. ISO C++ (Standard C++ Foundation) — Офіційний портал стандартів мови. [Електронний ресурс]. — Режим доступу: <https://isocpp.org/>
2. Cppreference.com — Повна довідкова система з мови C++ та бібліотеки STL. [Електронний ресурс]. — Режим доступу: <https://en.cppreference.com/>
3. LearnCpp.com — Спеціалізований навчальний ресурс із сучасного програмування на C++. [Електронний ресурс]. — Режим доступу: <https://www.learncpp.com/>
4. Refactoring.Guru — Керівництво з патернів проектування та рефакторингу (українська версія). [Електронний ресурс]. — Режим доступу: <https://refactoring.guru/uk>
5. Microsoft Learn (C++ Documentation) — Документація з розробки на Visual C++. [Електронний ресурс]. — Режим доступу: <https://learn.microsoft.com/en-us/cpp/>

6. Compiler Explorer (Godbolt) — Інструмент для інтерактивного аналізу компіляції коду. [Електронний ресурс]. — Режим доступу: <https://godbolt.org/>