

**ТАВРІЙСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ В. І. ВЕРНАДСЬКОГО**
Навчально-науковий інститут муніципального управління
та міського господарства
Кафедра комп'ютерних та інформаційних технологій



ЗАТВЕРДЖУЮ

Директор інституту
Володимир КИСЕЛЬОВ

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

першого (бакалаврського) рівня вищої освіти

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 122 «Комп'ютерні науки»

Освітньо–професійна програма: «Комп'ютерні науки»

Форма здобуття освіти: очна (денна)

КИЇВ 2025

УКЛАДАЧ силабусу к.т.н., доцент, доцент кафедри комп'ютерних та інформаційних технологій

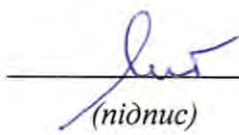
 Любов ОЛЕЩЕНКО
(підпис)

Розглянуто та схвалено на засіданні кафедри комп'ютерних та інформаційних технологій

Протокол № 1 від 26.08.2025

Завідувач кафедри  Олександр ГУЙДА
(підпис)

Гарант освітньо-професійної програми «Комп'ютерні науки»

к.т.н., доцент  Сергій ЛІСОВЕЦЬ
(підпис)

1. Загальна інформація про навчальну дисципліну	
1.Назва навчальної дисципліни, код в ОПП	Об'єктно-орієнтоване програмування ОК 1.2.19
2.Статус навчальної дисципліни	Навчальна дисципліна професійної підготовки
3.Рік навчання, семестр у якому викладається дисципліна	4 рік навчання, 7 та 8 семестри
4.Обсяг навчальної дисципліни (кількість кредитів, загальна кількість годин (аудиторних за видами занять, самостійної роботи здобувача вищої освіти)	7 кредитів (210 год.) загальна кількість годин: аудиторних: 76 лекцій: 26 год. практичні: 50 год. самостійна робота: 134 год.
5.Вид підсумкового (семестрового) контролю	Екзамен 8 семестр Залік 7 семестр
6.Інформація про консультації	Згідно затвердженого графіка консультацій
7.Мова викладання	українська
8.Прізвище, ім'я, по батькові викладача (науковий ступінь, вчене звання, посада)	К.т.н., доцент кафедри комп'ютерних та інформаційних технологій Олещенко Любов Михайлівна
9.Контактна інформація викладача	oleshchenko.liubov@tnu.edu.ua
10.Посилання на силабус на веб-сайті Університету	https://tnu.edu.ua
2. Опис навчальної дисципліни	
Анотація дисципліни	Навчальна дисципліна «Об'єктно-орієнтоване програмування» є складовою циклу професійної підготовки фахівців освітньо-кваліфікаційного рівня «бакалавр». Курс підпорядковано вирішенню таких основних задач, як здобуття теоретичних та практичних навичок з основ алгоритмізації та технології створення програм, вивчення синтаксису та опанування можливостей мови програмування Python.

Мета, завдання та цілі вивчення дисципліни	<u>Мета дисципліни</u> Метою викладання навчальної дисципліни «Об'єктно-орієнтоване програмування» є вивчення студентами принципів програмування на мові Python, отримання практичних навичок створення прикладних програм і реалізації алгоритмів обробки інформації на мові Python. Особлива увага приділяється розвитку практичних навиків програмування, аналізу та використанню алгоритмів і патернів програмування на мові Python. <u>Завдання і цілі дисципліни</u> Завданнями дисципліни є ознайомлення студентів з основними поняттями і принципами об'єктно-орієнтованих програмування на мові Python, отримання студентами практичних навичок реалізації задач автоматичної обробки інформації, написання комп'ютерних програм з використанням об'єктно-орієнтованих підходів програмування.
Пререквізити	Програмна інженерія, Системне програмування
Постреквізити	Підготовка до атестації здобувачів вищої освіти (бакалаврська кваліфікаційна робота).
Формат проведення дисципліни	Змішаний; У разі роботи в дистанційному режимі використовується корпоративне середовище Google Classroom; Лекції та практичні у дистанційному режимі проводяться через Google Meet; поточна комунікація з викладачем здійснюється через корпоративну пошту.

3. Перелік компетентностей, яких набувають здобувачі вищої освіти при вивченні навчальної дисципліни

ІК. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності

ЗК5. Здатність спілкуватися іноземною мовою

ЗК6. Здатність вчитися й оволодівати сучасними знаннями

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК9. Здатність працювати в команді

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт

СК2. Здатність до виявлення статистичних закономірностей недетермінованих явищ, застосування методів обчислювального інтелекту, зокрема статистичної, нейромережевої та нечіткої обробки даних, методів

машинного навчання та генетичного програмування тощо

СК6. Здатність до системного мислення, застосування методології системного аналізу для дослідження складних проблем різної природи, методів формалізації та розв'язування системних задач, що мають суперечливі цілі, невизначеності та ризики

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління

СК12. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення

СК13. Здатність до розробки мережевого програмного забезпечення, що функціонує на основі різних топологій структурованих кабельних систем, використовує комп'ютерні системи і мережі передачі даних та аналізує якість роботи комп'ютерних мереж

СК14. Здатність застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти й експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури

СК18. Здатність проектувати, впроваджувати, тестувати, налагоджувати, профілювати та оптимізувати критичні до швидкодії і часу виконання складові програмного забезпечення на мовах програмування низького і високого рівнів

СК19. Здатність розробляти та впроваджувати в міське господарство сучасні засоби комп'ютерної техніки для розв'язання прикладних задач управління в енергетичних, транспортних і санітарно-технічних підприємствах, ремонтно-експлуатаційних організаціях, шляховому господарстві тощо

4. Програмні результати навчання відповідно до освітньо-професійної програми

РН5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

РН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

PH13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення

PH14. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

5. Тематика та зміст навчальної дисципліни

Номер та назва розділу, теми, перелік основних питань	Вид навчального заняття	Форми і методи контролю знань	Кількість годин Лекція/ практичне заняття
Розділ 1. Основи об'єктно-орієнтованого програмування.			
Тема 1. Вступ до об'єктно-орієнтованого програмування в Python. Основні поняття об'єктно-орієнтованого програмування. 1. Основні принципи ООП: спадкування, інкапсуляція, поліморфізм. 2. Ініціалізація об'єктів. Метод <code>init</code> . 3. Перенавантаження операторів.	Лекція/ практичне заняття	усне опитування/виконання практичної роботи за темою	4/4
Тема 2. Класи та об'єкти. 1. Створення конструктора класу. 2. Створення класів і об'єктів. 3. Властивості та методи класів. 4. Створення об'єктів дати і часу.	Лекція/ практичне заняття	усне опитування/виконання практичної роботи за темою	4/4
Тема 3. Наслідування та композиція. 5. Використання наслідування для розширення функціональності 6. Роль композиції в ООП. 7. Діаграми класів.	Лекція/ практичне заняття	усне опитування/виконання практичної роботи за темою	2/6

Тема 4. Поліморфізм та інкапсуляція. 1. Різновиди поліморфізму в Python. 2. Використання поліморфізму для підвищення гнучкості програм. 3. Інкапсуляція. 4. Використання абстракції для спрощення складних систем 5. Створення інтерфейсів для взаємодії між класами.	Лекція/ практичне заняття	усне опитування/ <i>виконання практичної роботи за темою</i>	4/6
Розділ 2. Додаткові можливості ООП в Python.			
Тема 5. Спеціальні методи класу. 1. Роль спеціальних методів у Python. 2. Перевизначення різних спеціальних методів для власних потреб.	Лекція/ практичне заняття	усне опитування/ <i>виконання практичної роботи за темою</i>	2/6
Тема 6. Декоратори класів та методів. 1. Використання декораторів для зміни поведінки класів та методів. 2. Практичне застосування декораторів у програмуванні на Python.	Лекція/ практичне заняття	усне опитування/ <i>виконання практичної роботи за темою</i>	2/6
Розділ 3. Використання пакетів та модулів у ООП.			
Тема 7. Використання пакетів та модулів у ООП. 1. Імпортування класів та об'єктів з інших файлів. 2. Організація проєкту з використанням модулів.	Лекція/ практичне заняття	усне опитування/ <i>виконання практичної роботи за темою</i>	4/6

Тема 8. Обробка файлів. 1. Класи та об'єкти для роботи з файлами: інкапсуляція файлових операцій. 2. Режими відкриття файлів і керування життєвим циклом файлового об'єкта. 3. Реалізація читання та запису даних у файли методами класів. 4. Абстракція файлового доступу та повторне використання коду.	Лекція/ практичне заняття	усне опитування/ <i>виконання практичної роботи за темою</i>	2/6
Тема 9. Робота з виключеннями. 1. Поняття виключень та їх роль у забезпеченні надійності програмного забезпечення. 2. Класи виключень, ієрархія виключень та механізм їх поширення 3. Обробка виключень. Використання власних виключень для підвищення керованості та читабельності коду.	Лекція/ практичне заняття	усне опитування/ <i>виконання практичної роботи за темою</i>	2/6

6. Інформація про індивідуальне завдання

Курс дисципліни «Об'єктно-орієнтоване програмування» передбачає виконання індивідуальних завдань у вигляді самостійної роботи. Провідна мета організації самостійної роботи полягає у необхідності широкого огляду тематики курсу з використанням основної та додаткової літератури, набуття навичок пошуку необхідної інформації, її аналітичного осмислення. В процесі цієї роботи студенти повинні навчитися робити узагальнюючі висновки, оформляти результати роботи та планувати свою діяльність з вивчення дисципліни. Контроль за самостійною роботою студентів – поточний контроль, тестування, контрольна робота. Завдання контрольної роботи для студентів заочної форми навчання містять індивідуальні завдання для кожного студента.

7. Технічне обладнання та програмне забезпечення

У звичайному режимі навчання вивчення навчальної дисципліни передбачає приєднання кожного здобувача до навчального середовища Google Classroom, оскільки там розміщуються навчальні матеріали, проводиться тестування, ведеться журнал оцінювання навчальних досягнень.

У режимі дистанційного навчання вивчення курсу додатково передбачає приєднання кожного здобувача вищої освіти до програм Google Meet (для занять у режимі відеоконференцій).

Для комунікації та опитувань, виконання домашніх завдань, виконання завдань самостійної роботи, проходження тестування (поточний, підсумковий контроль) тощо, здобувачу пропонується самостійно потурбуватися про якість доступу до мережі Інтернет, наявність ноутбука або персонального комп'ютера, мобільного пристрою (телефона, планшета) з підключенням до мережі Інтернет.

8. Політика дисципліни.

Політика щодо академічної доброчесності: списування під час тесту, екзамену заборонені.

Жодні форми порушення академічної доброчесності не толеруються. У випадку таких подій – реагування відповідно до Методичних рекомендацій для закладів вищої освіти з підтримки принципів академічної доброчесності.

Правила перезарахування кредитів у випадку мобільності, правила перескладання або відпрацювання пропущених занять тощо: відбувається згідно з Положення про організацію освітнього процесу у Таврійському національному університеті ім. В.І. Вернадського.

Політика щодо дедлайнів та перескладання: роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (до - 50% від можливої максимальної кількості балів за вид діяльності).

Перескладання тесту відбувається за наявності поважних причин (наприклад, лікарняний).

Політика щодо відвідування: відвідування занять є обов'язковим компонентом. За об'єктивних причин (наприклад, хвороба, працевлаштування, міжнародне стажування) навчання може відбуватись в онлайн-формі за погодженням.

9. Система оцінювання та вимоги

З дисципліни ЗВО (здобувач вищої освіти) може набрати до 60% підсумкової оцінки за виконання всіх видів робіт, що виконуються протягом семестру і до 40% підсумкової оцінки – при виконанні екзаменаційної роботи.

Поточний контроль проводиться шляхом спілкування із ЗВО під час лекцій, консультацій та опитувань.

Результати поточного контролю за відповідний модуль оприлюднюються викладачем на наступному аудиторному занятті. Бали, які набрані ЗВО під час аудиторних занять, складають оцінку поточного контролю.

Семестровий контроль у вигляді екзаменаційної роботи проводиться під час сесії з трьома теоретичними питаннями та двома практичними завданнями

(40 балів максимум). Оцінка за результатами вивчення дисципліни формується шляхом додавання підсумкових результатів поточного контролю до екзаменаційної оцінки. Взаємозв'язок між набраними балами і оцінкою наведено у розділі 9.1. У випадку, якщо ЗВО протягом семестру не виконав в повному обсязі передбачених робочою програмою всіх видів навчальної роботи, має невідпрацьовані роботи або не набрав мінімально необхідну кількість балів (20), він не допускається до складання екзамену під час сесії, але має право ліквідувати академічну заборгованість. Повторне складання екзамену з метою підвищення позитивної оцінки не дозволяється.

9.1.Шкала та схема формування підсумкової оцінки

Теми			Сума
Розділ 1	Розділ 2	Розділ 3	60 балів
20 балів	20 балів	20 балів	
Підсумковий контроль			40 балів
Максимальна сума балів			100 балів

Шкала оцінювання: національна та ECTS

Оцінка в балах	Оцінка за національною шкалою	Оцінка за шкалою ECTS	
		Оцінка за шкалою ECTS	Пояснення
90-100	відмінно	A	Відмінно-відмінний рівень знань (умінь) в межах обов'язкового матеріалу з можливими незначними недоліками
80-89	добре	B	Дуже добре - рівень знань достатньо високий (умінь). В межах обов'язкового матеріалу без суттєвих (грубих) помилок
75-79	добре	C	Добре - в цілому добрий рівень знань (умінь) з незначною кількістю помило
66-74	задовільно	D	Задовільно - посередній рівень знань (умінь) із значною кількістю недоліків, достатній для подальшого навчання або професійної діяльності
60-65	задовільно	E	Достатньо мінімально можливий допустимий рівень знань (умінь)
30-59	незадовільно	FX	Незадовільно - потрібно додатково працювати для отримання позитивної оцінки
0-29	незадовільно	F	Незадовільно - необхідна серйозна подальша робота з повторним вивченням курсу

Загальна система оцінювання курсу	Підсумкова оцінка з дисципліни є сумою оцінок з відповідною вагою за кожен з таких видів робіт: робота на практичних заняттях та підсумковий контроль (екзамен). Підсумкова оцінка визначається відповідно до таблиці оцінювання за різними шкалами (100-бальна, ECTS, національна).
Розрахункова графічна-робота	В рамках курсу не передбачено виконання РГР.
Лабораторні та практичні роботи	Критерії оцінювання лабораторних та практичних робіт: 1. Підготовленість до практичних занять відповідно до лекційного курсу. 2. Самостійність виконання практичних робіт. 3. Повнота виконання завдань. 4. Своєчасність виконання та захисту практичних робіт. Максимальний бал за кожен практичну роботу – 10 балів.
Тест	Проміжний тест проводиться по кожному розділу курсу та оцінюється максимально в 10 балів.
Екзамен	Екзамен проводиться в кінці курсу, включає три теоретичні питання і два практичні завдання. Максимально оцінюється в 40 балів.
Умови допуску до підсумкового контролю	Позитивна оцінка за всіма обов'язковими видами робіт (практичні роботи).

10. Рекомендовані джерела інформації

Назва теми	Рекомендовані джерела інформації до теми (основна література; допоміжна література; інформаційні ресурси в мережі Інтернет)
Розділ 1. Основи об'єктно-орієнтованого програмування.	

Тема 1. Вступ до об'єктно-орієнтованого програмування в Python. Тема 2. Класи та об'єкти. Тема 3. Наслідування та композиція. Тема 4. Поліморфізм та інкапсуляція.	Основна література: 1 - стор.8-24, 83-85 2 - стор. 5-25, 45. Додаткова література: 1, 2, 3, 4.
Розділ 2. Додаткові можливості ООП в Python.	
Тема 5. Спеціальні методи класу. Тема 6. Декоратори класів та методів.	Основна література: 1 - стор. 89-92, 2 - стор. 144-150. Додаткова література: 5.
Розділ 3. Використання пакетів та модулів у ООП.	
Тема 7. Використання пакетів та модулів у ООП. Тема 8. Обробка файлів. Тема 9. Робота з виключеннями.	Додаткова література: 6, 7, 8.

Основна література

1. Об'єктно-орієнтоване програмування мовою Python: Конспект лекцій: навч. посіб. / КПІ ім. Ігоря Сікорського; уклад.: Д. Д. Татарчук, Ю. В. Діденко. –Київ: КПІ ім. Ігоря Сікорського, 2021. –129 с.
2. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник – К.: ВПЦ "Київський Університет", 2020. – 152 с.

Додаткова література

1. Класи та об'єкти в Python [Електронний ресурс]: <https://docs.python.org/3/tutorial/classes.html>
2. Наслідування в Python [Електронний ресурс]: <https://ioflood.com/blog/python-inheritance/>

3. Інкапсуляція в Python [Електронний ресурс]:
https://www.tutorialspoint.com/python/python_encapsulation.htm
4. Поліморфізм в Python [Електронний ресурс]:
<https://www.javatpoint.com/polymorphism-in-python>
5. Декоратори в Python [Електронний ресурс]: <https://acode.com.ua/decorators-python/>
6. Пакети в Python [Електронний ресурс]: <https://acode.com.ua/packages-python/>
7. Робота з файлами в Python [Електронний ресурс]: <https://acode.com.ua/file-operations-python/>
8. Обробка винятків [Електронний ресурс]: <https://docs.python.org/uk/3.13/c-api/exceptions.html>