



Кафедра загальної педагогіки та права

Схвалено:

На засіданні кафедри загальної педагогіки та права

Протокол №10 від 28 травня 2024 р.

Завідувач кафедри _____ Вікторія ПРИЛИПКО

**СИЛАБУС
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ОСНОВИ ПРОГРАМУВАННЯ ТА ОБ'ЄКТНО-ОРІЄНТОВНЕ
ПРОГРАМУВАННЯ»**

Спеціальності: 015 Професійна освіта (за спеціалізаціями)

Галузь знань: 01 Освіта

Освітнього рівня: перший (бакалаврський) рівень

Освітня кваліфікація: Бакалавр з професійної освіти за спеціалізацією «Цифрові технології»

Спеціалізація: цифрові технології



Розробники силябусу навчальної дисципліни:

Кришан Олексій Федорович – кандидат економічних наук, доцент кафедри інформаційних технологій

(підпис)

Викладач:

Кришан Олексій Федорович – кандидат економічних наук, доцент кафедри інформаційних технологій

(підпис)

Силябус розглянуто на засіданні кафедри загальної педагогіки та права

Протокол № 10 від « 28 » травня 2024р.

1. Загальна інформація про навчальну дисципліну

Назва навчальної дисципліни	Основи програмування та об'єктно-орієнтовне програмування
Шифр та назва спеціальності	015 Професійна освіта (цифрові технології)
Рівень вищої освіти	перший (бакалаврський) рівень
Статус дисципліни	обов'язкова
Кількість кредитів і годин	14 кредити / 420 год Лекції: 84 Практичні заняття: 86 Самостійна робота студентів: 164
Терміни вивчення дисципліни	2 - 3 семестри
Мова викладання	українська
Вид підсумкового контролю	II семестр - Залік III семестр - Екзамен

2. Загальна інформація про викладача. Контактна інформація

Кришан Олексій Федорович	
Науковий ступінь	кандидат економічних наук
Вчене звання	доцент
Посада	доцент кафедри загальної педагогіки та права
Контактна інформація викладача:	
E-mail	krishan@ki-maup.com.ua
Контактний тел.:	+380677399694

3. Анотація курсу

Вивчення навчальної дисципліни «Основи програмування та об'єктно-орієнтовне програмування» є фундаментальною складовою підготовки майбутнього фахівця у сфері інформаційних технологій. Він послідовно занурює здобувачів освіти у світ програмування, починаючи з базових концепцій алгоритмізації та структурного підходу, і завершуючи потужною парадигмою об'єктно-орієнтованого програмування (ООП). На прикладі сучасної мови програмування Python студенти пройдуть шлях від написання перших скриптів до проєктування складних, гнучких та масштабованих програмних систем. Курс робить акцент на активному навчанні та виробленні практичних навичок розробки, тестування та налагодження програм.

Предметом вивчення навчальної дисципліни є методи та технології розробки алгоритмів, базові та складні структури даних, принципи процедурного програмування, а також ключові концепції об'єктно-орієнтованої парадигми: інкапсуляція, спадкування та поліморфізм.

Метою курсу є формування у здобувачів вищої освіти цілісної системи теоретичних знань та практичних навичок, необхідних для розв'язання прикладних задач за допомогою комп'ютера. Мета полягає у розвитку алгоритмічного та об'єктно-орієнтованого мислення, що дозволить свідомо обирати та застосовувати ефективні підходи до проєктування, розробки та супроводу програмного забезпечення

Пререквізити та постреквізити

Пререквізити:

Вивчення дисципліни Основи програмування та об'єктно-орієнтовне програмування базується на знаннях, отриманих у рамках шкільного курсу інформатики та математики.

Постреквізити:

Знання та вміння, отримані в процесі вивчення курсу, є основою для успішного опанування таких дисциплін професійного циклу: «Робототехніка», «Інформаційні та комунікаційні технології в професійній освіті».

4. Програмні компетентності та результати навчання

Навчальна дисципліна забезпечує формування студентами загальних і спеціальних компетентностей та набуття результатів навчання, визначених освітньо-професійною програмою «Професійна освіта (цифрові технології)».

Загальні компетентності	K05. Здатність вчитися і оволодівати сучасними знаннями.
	K06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
Спеціальні компетентності	K16. Здатність використовувати сучасні інформаційні технології та спеціалізоване програмне забезпечення та інтегрувати їх в освітнє середовище.
	K19. Здатність використовувати відповідне програмне забезпечення для вирішення професійних завдань, відповідно до спеціалізації.
	K25. Здатність збирати, аналізувати та інтерпретувати інформацію (дані) відповідно до спеціалізації
Результати навчання	ПР 09. Відшуковувати, обробляти, аналізувати та оцінювати інформацію, що стосується професійної діяльності, користуватися спеціалізованим програмним забезпеченням та сучасними засобами зберігання та обробки інформації
	ПР 18. Розв'язувати типові спеціалізовані задачі, пов'язані з вибором матеріалів, виконанням необхідних розрахунків, конструюванням, проектуванням технічних об'єктів у предметній галузі (відповідно до спеціалізації).
	ПР 19. Уміти обирати і застосовувати необхідне устаткування, інструменти та методи для вирішення типових складних завдань у галузі (відповідно до спеціалізації)
	ПР 27*. Знати принципи й алгоритми використання цифрових технологій в освітній системі, володіти сучасними методами розробки цифрового освітнього середовища, уміти планувати, організовувати та контролювати роботу інформаційних систем та застосовувати їх з метою розробки програмних продуктів для закладів освіти

5. Зміст навчальної дисципліни та розподіл бюджету часу**2 СЕМЕСТР (Основи програмування)****Модуль 1. Основи алгоритмізації та мови програмування****Тема 1. Вступ до програмування. Парадигми програмування.**

історія мов програмування, компіляція та інтерпретація, огляд процедурної, об'єктно-орієнтованої та функціональної парадигм, переваги Python.

Тема 2. Алгоритми, їх властивості та способи опису.

визначення алгоритму, властивості дискретності та детермінованості, способи представлення (словесний, псевдокод, блок-схеми), базові алгоритмічні структури.

Тема 3. Основи Python: змінні, базові типи даних, оператори.

іменування змінних, динамічна типізація, цілі числа (int), числа з плаваючою точкою (float), логічний тип (bool), арифметичні та логічні оператори.

Тема 4. Введення та виведення даних. Форматування.

робота з функціями `input()` та `print()`, роздільники та кінцеві символи, сучасні методи форматування рядків (f-strings, метод `.format()`).

Тема 5. Умовні конструкції: `if`, `else`, `elif`.

логіка розгалуження в програмах, вкладені умови, складні логічні вирази (`and`, `or`, `not`), тернарний оператор.

Тема 6. Цикли: `for` та `while`. Керування ітераціями.

ітерація по послідовностях, робота з функцією `range()`, нескінченні цикли, оператори переривання `break`, `continue` та блок `else` в циклах.

Тема 7. Модульний контроль №1.

перевірка теоретичних знань та практичних навичок з основ синтаксису, алгоритмізації та керуючих конструкцій мови Python.

Модуль 2. Структура та аналітика даних. Процедурне програмування. Хмарні сервіси.**Тема 8. Робота з рядками та їх методи.**

рядки як незмінні послідовності, зрізи (slices), основні методи обробки тексту (`upper`, `lower`, `find`, `replace`, `split`, `join`).

Тема 9. Списки (Lists) як динамічні масиви.

створення списків, індексація, додавання та видалення елементів, методи `sort` та `reverse`, спискові включення (list comprehensions).

Тема 10. Кортежі (Tuples) та словники (Dictionaries).

незмінні структури даних, робота з парами «ключ-значення», методи словників, ефективність пошуку даних, множини (sets).

Тема 11. Функції. Основи процедурного програмування.

визначення та виклик функцій, оператор `return`, документування коду (docstrings), декомпозиція задач на підзадачі.

Тема 12. Області видимості. Передача параметрів у функції.

локальні та глобальні змінні, позиційні та іменовані аргументи, значення за замовчуванням, робота з `*args` та `**kwargs`.

Тема 13. Робота з файлами.

відкриття файлів (`open`), режими доступу (`read`, `write`, `append`), менеджер контексту `with`, обробка текстових даних, захист індивідуального проєкту.

ІНДЗ. Виконання та захист індивідуального навчально-дослідного завдання.

3 СЕМЕСТР (Об'єктно-орієнтоване програмування)**Модуль 3. Парадигма ООП та базові концепції****Тема 14. Вступ до ООП. Поняття класу та об'єкту.**

об'єктна модель світу, клас як креслення та об'єкт як його екземпляр, атрибути класу та атрибути екземпляра.

Тема 15. Конструктори та деструктори. Вказівник `self`.

ініціалізація об'єкта через метод `_init_`, призначення `self` як посилання на поточний об'єкт, керування життєвим циклом об'єкта.

Тема 16. Інкапсуляція. Специфікатори доступу (public, protected, private).

приховування внутрішньої логіки, угода про найменування (приватні та захищені атрибути), методи доступу, захист даних від некоректних змін.

Тема 17. Властивості (property). Getters & Setters.

використання декоратора @property, створення обчислювальних атрибутів, валідація даних при записі в поля класу.

Тема 18. Модульний контроль №2.

оцінювання навичок процедурного програмування, роботи зі структурами даних та базових концепцій створення класів.

Модуль 4. Механізми спадкування та поліморфізму**Тема 19. Спадкування. Ієрархія класів.**

повторне використання коду, базові та похідні класи, розширення функціоналу батьківського класу, використання функції super().

Тема 20. Множинне спадкування та проблема ромба (MRO).

особливості спадкування від кількох класів, лінеаризація класів, алгоритм пошуку методів (Method Resolution Order).

Тема 21. Поліморфізм. Перевизначення методів.

однаковий інтерфейс для різних типів об'єктів, качина типізація (duck typing), зміна поведінки батьківських методів у нащадках.

Тема 22. Абстрактні класи та методи.

використання модуля abc, створення інтерфейсів, заборона створення екземплярів базового класу, обов'язкова реалізація методів у підкласах.

Тема 23. Модульний контроль №3.

перевірка володіння основними принципами ООП (спадкування, поліморфізм) та архітектурного мислення

Модуль 5. Розширені можливості ООП та патерни**Тема 24. Перевантаження операторів.**

магічні методи (_add_, _str_, _len_), адаптація поведінки стандартних операторів (+, -, ==) для користувацьких об'єктів.

Тема 25. Обробка виняткових ситуацій (try-except-finally).

типи помилок, створення власних класів винятків, безпечне завершення програм, логування помилок.

Тема 26. Вступ до патернів проєктування (Singleton, Factory Method).

концепція шаблонів, реалізація Singleton (одинак) для єдиного доступу до ресурсу, Factory Method (фабричний метод) для гнучкого створення об'єктів.

Таблиця бюджету навчального часу (год)

Тема / завдання	Бюджет навчального часу (год)		
	Лекції	Практичні (семінарські) заняття	Самостійна робота студента



II семестр			
Модуль 1. Основи алгоритмізації та мови програмування			
Тема 1.	2	2	10
Тема 2.	4	2	6
Тема 3.	4	4	10
Тема 4.	2	4	8
Тема 5.	4	4	8
Тема 6.	2	2	8
Тема 7.	2	2	6
Підсумковий модульний контроль		1	2
Модуль 2. Структури даних та процедурне програмування			
Тема 8.	4	4	8
Тема 9.	4	4	10
Тема 10.	4	4	8
Тема 11.	4	4	10
Тема 12.	2	2	10
Тема 13.	4	2	10
Підсумковий модульний контроль		1	2
Всього за семестр:	42	42	116
III семестр			
Модуль 3. Парадигма ООП та базові концепції			
Тема 14.	2	4	8
Тема 15.	4	4	10
Тема 16.	4	4	10
Тема 17.	4	4	10
Тема 18.	2	2	10
Підсумковий модульний контроль		1	3
Модуль 4. Механізми спадкування та поліморфізму			
Тема 19.	2	4	10
Тема 20.	4	4	10
Тема 21.	4	4	10
Тема 22.	2	2	10
Тема 23.	4	2	8
Підсумковий модульний контроль		1	3
Модуль 5. Розширені можливості ООП та патерни			
Тема 24.	4	4	10
Тема 25.	4	2	10
Тема 26.	2	1	8
Підсумковий модульний контроль		1	4
Всього за семестр:	42	44	134
Всього:	84	86	250

6. Методи навчання та система оцінювання

Навчальний процес організовано з акцентом на активну позицію здобувачів освіти.

Лекційні та практичні заняття проводяться у аудиторіях Інституту та/або з використанням технологій дистанційного навчання на платформі Zoom за допомогою поєднання різних форм та методів навчання.

Поточний контроль здійснюється за результатами усіх виконаних завдань, передбачених політикою оцінювання.

Тестування проводиться з використанням автоматизованих систем онлайн-тестування в синхронному або асинхронному режимах. Кількість балів за тест визначається в залежності від критеріїв (складності питання, наявності декількох правильних відповідей на питання тощо).

Робота на практичних заняттях передбачає опрацювання лекційного матеріалу з поєднанням таких форм і методів навчання, як тестування, кейс-завдання, практичні завдання (робота з файлами) тощо.

Індивідуальні навчально-дослідні завдання (ІНДЗ) мають за мету сформувати у студентів навички самостійного дослідження та здатності застосовувати набуті знання у практичних ситуаціях. ІНДЗ — це імітація реального робочого процесу, це перший крок студента до того, щоб думати і діяти як інженер програмного забезпечення.

Відвідування практичних занять є обов'язковим. Відпрацювання пропущених практичних занять відбувається в індивідуальному порядку під час консультацій (в очному режимі або в режимі on-line на платформі Zoom).

Система оцінювання та вимоги

Програмування — це насамперед практика. Тому система оцінювання побудована так, щоб максимально заохочувати активну роботу студентів протягом усього курсу. Ключова мета — не просто «здати і забути», а набути реальних навичок, які стануть фундаментом для майбутньої кар'єри програміста.

Контроль успішності здобувачів освіти поділяється на поточний і підсумковий (семестровий).

Поточний контроль здійснюють під час проведення практичних занять, метою якого є систематична перевірка розуміння та засвоєння теоретичного навчального матеріалу, уміння використовувати теоретичні знання під час виконання практичних завдань тощо.

Поточний контроль здійснюється за результатами усіх виконаних завдань, передбачених політикою оцінювання.

Модульний контроль. Модульна контрольна робота з навчального курсу проводиться на останньому занятті модуля у вигляді тестування, а саме, тести закритої форми: тест-альтернатива, тест-відповідність.

Самостійна робота студентів при вивченні дисципліни Основи програмування та об'єктно-орієнтовне програмування є невід'ємною частиною освітнього процесу, спрямованою на поглиблення, узагальнення та закріплення знань. Вона включає такі основні види діяльності:

Підготовка до аудиторних занять: Це передбачає опрацювання матеріалів пройдених лекцій для якісної роботи на наступних практичних заняттях.

Робота з інформаційними джерелами: Студенти мають вивчати базову і додаткову літературу, технічну документацію, періодичні видання та професійні Інтернет-ресурси.

Виконання практичних завдань: Розробка програмного коду, вирішення алгоритмічних задач та налагодження програм, що доповнюють та розширюють теми, розглянуті на практичних заняттях.

Самостійне опрацювання тем: Вивчення окремих тем або додаткових аспектів дисципліни, які винесені на самостійне опрацювання згідно з навчальною програмою.

Підготовка до контрольних заходів: Систематична підготовка до всіх видів поточного та підсумкового контролю, включаючи модульні контрольні роботи, залік та іспит.

Проектна діяльність: Виконання індивідуальних або колективних навчальних проєктів, що передбачають застосування отриманих знань для вирішення практичних завдань у конкретних предметних областях.

Науково-дослідна робота (за бажанням): Участь у роботі наукових гуртків, підготовка доповідей для студентських конференцій, написання тез або статей.

Індивідуальні навчально-дослідні завдання (ІНДЗ) мають за мету сформувати у студентів навички самостійних досліджень й здатність застосовувати набуті знання у практичних ситуаціях.

ІНДЗ — це особистий «тест-драйв» студентів в умовах, наближених до реальної роботи. На лекціях та практичних студенти отримують інструменти (знання мови, алгоритмів), а ІНДЗ — це проєкт, де вони повинні самостійно збудувати щось працююче з цих інструментів.

Насправді це перевірка найважливіших якостей майбутнього програміста:

Вміння вчитися самостійно. При виконанні завдання студент буде стикатися з новими бібліотеками, невідомими помилками та нестандартними задачами. ІНДЗ навчить шукати інформацію, читати документацію та знаходити рішення в таких ресурсах, як Stack Overflow чи офіційна документація Python.

Перехід від теорії до практики. Вивчаючи дисципліну студент при виконанні завдання повинен вміти створити цілісний програмний проєкт, який працює без помилок, коректно обробляє різні дані та має логічну структуру.

Прийняття інженерних рішень. Студент сам обирає, які структури даних використати, як організувати код, як розбити велике завдання на менші частини. Це розвиває алгоритмічне та логічне мислення.

Відповідальність за результат. Якість, функціональність та навіть читабельність коду безпосередньо впливають на оцінку. Це привчає до стандартів якості, які є обов'язковими у професійній розробці.

Таким чином, ІНДЗ — це не просто ще одна контрольна. Це імітація реального робочого процесу, яка перетворює слухача курсу на початківця-розробника. Це перший крок студента до того, щоб думати і діяти як інженер програмного забезпечення.

ІНДЗ оцінюється у **20 балів** за наступними критеріями:

16-20 балів (Відмінно) Студент виконав програмний проєкт, який повністю відповідає поставленому завданню.

Якість коду: Програмний код є ефективним, добре структурованим, читабельним та містить необхідні коментарі. Використані доцільні алгоритми та структури даних.

Функціональність: Програма працює без помилок, коректно обробляє стандартні та граничні випадки (edge cases).

Захист роботи: Студент демонструє глибоке розуміння власного коду, може чітко пояснити логіку роботи програми, обґрунтовує вибір використаних методів та підходів. Впевнено та повно відповідає на всі питання викладача.

10 - 15 балів (Добре) Студент виконав програмний проєкт, який вирішує основну частину завдання, але має певні недоліки.

Якість коду: програмний код є функціональним, але може бути неоптимальним, недостатньо структурованим або мати несуттєві стилістичні помилки. Коментарі часткові або не повністю розкривають логіку.

Функціональність: програма виконує основні функції, але може містити незначні помилки або некоректно обробляти деякі вхідні дані.

Захист роботи: студент загалом розуміє, як працює його програма, але не може детально пояснити окремі фрагменти коду або обґрунтувати свій вибір. На питання викладача відповідає невпевнено або потребує уточнень та допомоги.

5 - 9 балів (Задовільно) Студент надав роботу, яка лише частково вирішує поставлене завдання або містить суттєві помилки.

Якість коду: код неструктурований, важкий для розуміння, містить грубі логічні помилки. Використані неефективні або застарілі підходи.

Функціональність: програма не працює належним чином, не виконує більшість із заявлених вимог або аварійно завершує роботу.

Захист роботи: студент не може пояснити принципи роботи власної програми та логіку її виконання. Не може відповісти на питання викладача навіть із його допомогою.

Таблиця розподілу балів (2-й семестр, Залік)



Тема/завдання	Робота на практич./семінар. занятті	Самостійна робота	ІНДЗ (за потреби)	Всього
Модуль І				
Тема 1.	3	1		4
Тема 2.	3	1		4
Тема 3.	4	1		5
Тема 4.	4	1		5
Тема 5.	4	1		5
Тема 6.	4	1		5
Тема 7.	4	1		5
Модульна контрольна робота	7			7
Разом за І модуль	33	7		40
Модуль ІІ				
Тема 8.	5	1		6
Тема 9.	6	1		7
Тема 10	6	1		7
Тема 11	6	1		7
Тема 12	6	1		7
Тема 13	5	1		6
ІНДЗ			20	20
Разом за ІІ модуль	34	6	20	60
Загальна кількість балів за семестр				100

Підсумковий контроль проводиться у формі заліку. Залік виставляється за результатами роботи здобувача впродовж семестру. Студенти, які набрали 60 і більше балів, отримують «зараховано» автоматично. Студенти, що набрали менше 60 балів, або бажають підвищити оцінку, складають залік в усній або письмовій формі на останньому занятті.

Таблиця розподілу балів (3-й семестр, Екзамен)

Тема/завдання	Робота на практич./семінар. занятті	Самостійна робота	Іспит	Всього
Модуль І				
Тема 14.	1	1		2
Тема 15.	2	1		3
Тема 16.	3	1		4
Тема 17.	4			4
Тема 18.	3	1		4
Модульна контрольна робота	5			5
Разом за І модуль	18	4		22
Модуль ІІ				
Тема 19.	1			1
Тема 20	2	1		3
Тема 21	4	1		5



Тема 22	4			4
Тема 23	4	1		5
Модульна контрольна робота	5			5
Разом за II модуль	20	3		23
Модуль III				
Тема 24	4	1		2
Тема 25	4	1		3
Тема 26	4	1		3
Разом за III модуль	12	3		15
Іспит			40	40
Загальна кількість балів				100

До екзамену допускаються студенти, які виконали усі види робіт, передбачені навчальним планом. Підсумкова оцінка з дисципліни складається з двох складових: стартової (заходи поточного контролю, максимум 60 балів) та екзаменаційної (завдання на іспиті, максимум 40 балів).

7. Технічне обладнання та/або програмне забезпечення

В освітньому процесі використовуються комп'ютерні класи, мультимедійне обладнання для проведення лекційних та практичних занять. Для виконання завдань студенти використовують мову програмування

Python та стандартне середовище розробки (IDE), рекомендоване кафедрою. Необхідний доступ до мережі Інтернет для роботи з документацією та онлайн-ресурсами, що забезпечується в межах інституту

IDE (Інтегроване середовище розробки): Visual Studio 2022, JetBrains Rider, CLion, або Visual Studio Code з відповідними плагінами для C++.

Компілятор: MSVC (входить до Visual Studio), GCC/G++ (MinGW), Clang.

Система контролю версій: Git, GitHub/GitLab.

Програмне забезпечення для моделювання: UML-редактори (напр., StarUML, diagrams.net).

8. Політика курсу:

Для успішного засвоєння курсу здобувач освіти має регулярно відвідувати лекційні та практичні заняття, активно працювати, своєчасно та якісно виконувати всі завдання для самостійної роботи та практичні завдання. Пропущені заняття мають бути відпрацьовані у встановленому порядку. Курс передбачає дотримання принципів **академічної доброчесності**. Усі роботи (практичні, самостійні, контрольні) мають бути виконані самостійно. Виявлення плагиату, списування, фабрикації чи фальсифікації даних призведе до анулювання роботи з оцінкою «0».

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 - 100	A	відмінно	зараховано
82 - 89	B	добре	
75 - 81	C	задовільно	
68 - 74	D		
60 - 67	E		



35 - 59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
1 - 34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

9. Методичне забезпечення навчальної дисципліни.

Навчально-методичний супровід дисципліни це сукупність усіх навчальних і методичних матеріалів, які використовуються для викладання курсу.

10. Рекомендовані джерела

Основна:

1. Кришан О.Ф., Корень О.М. Навчально-методичний посібник «Основи програмування» Електронне видання. Кременчук, ПІ МАУП, 2025 р. -283 с
2. Matthes, E. (2023). *Python Crash Course, 3rd Edition: A Hands-On, Project-Based Introduction to Programming*. No Starch Press., 552 С. <https://balka-book.com/ua/python-70/python-crash-course-3rd-edition-a-handson-projectbased-introduction-to-programming-3rd-edition-259260>
3. Бандоріна Л.М. Основи алгоритмізації та програмування:навч. посібник / Л.М.Бандоріна, Т.О.Климович, К.О.Удача. УДУНТ, 2022. –158
<https://crust.ust.edu.ua/server/api/core/bitstreams/bfb9f823-172b-49bf-92a2-48bbbef5f113/content>
4. Баран С.В. Основи web-програмування: Навчальний посібник. – Кривий Ріг: Державний університет економіки і технологій, 2023. –316 с.
5. Бхаргава А. Грокаємо алгоритми. Ілюстрований посібник для програмістів і допитливих / пер. з англ. Київ : ArtHuss, 2023. – 256 с. <https://nashformat.ua/products/grokaemo-algorytmy-ilyustrovanyj-posibnyk-dlya-programistiv-i-dopytlyvyh-943384?srsId=AfmBOoqRuFb37NOxbrh0X3HmtvoT52WFgcgOKpVc421CucL3wy6sTUeT>.
6. Григоренко, Я. (2021). *Python. Швидкий старт*.
<https://edu.cbsystematics.com/ua/webinars/python-quick-start>
7. Григорович В. Г. Об'єктно-орієнтоване програмування. Частина 1: Основні поняття. – Львів: «Магнолія 2006», 2024. – 284 с. https://magnolia.lviv.ua/wp-content/uploads/2024/01/Zmist_Obiektno-orientovane-prohramuvannia.-CHastyna-1.pdf
8. Гришанович Т. О., Глинчук Л. Я. Основи об'єктно-орієнтованого програмування: навч. посібник. – Луцьк: ВНУ імені Лесі Українки, 2022. – 120 с. <https://evnuir.vnu.edu.ua/handle/123456789/20320>
9. Інформатика. Лабораторний практикум [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Інжиніринг зварювання, лазерних та споріднених технологій» спец. 131 «Прикладна механіка» / С. М. Мінаков, А.В. Чорний, А.С. Мінаков, І.М. Вдовіченко, Д.М. Вдовіченко; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2023.
10. Мнушка, О. В. Об'єктно-орієнтоване програмування мовою Python : навч. посіб. / Мнушка О. В., Савченко В. М., Маций О. Б. – Харків : ХНАДУ, 2021. – 228 с. <https://dspace.khadi.kharkov.ua/items/d2dcd029-4963-4223-b4f0-82eb01bcbfe4>
11. Муляр В. П. Об'єктно-орієнтоване програмування: лабораторний практикум. – Луцьк: Вежа-Друк, 2022. – 112 с. <https://evnuir.vnu.edu.ua/handle/123456789/21291>
12. Об'єктно-орієнтоване програмування. Лабораторний практикум: навчальний посібник / Л.Л. Омельчук, А.С. Белова. – Київ: 2022. - 271 с. https://csc.knu.ua/media/filer_public/64/78/64780f54-87c0-439b-8aaa-2afcff3e2b29/ooop_2022_print_20_09.pdf
13. Об'єктно-орієнтоване програмування. Практикум [Електронний ресурс] : навчальний посібник / КПІ ім. Ігоря Сікорського ; уклад. В. А. Данілова. – Електронні текстові дані – Київ : КПІ ім. Ігоря Сікорського, 2021. – 121 с. <https://ela.kpi.ua/items/d9760ade-916b-449f-9a7e-a8c1cf42bf14>

14. Основи веб-програмування. Курс JavaScript / W3Schools.ua, 2024.
URL: <https://w3schoolsua.github.io/js/index.html>
15. Основи програмування на C++: Навчальний посібник для студентів спеціальностей 113 – Прикладна математика та 122 – Комп'ютерні науки: навч. посіб./ Водка О.О., Дашкевич А.О., Іванченко К.В., Розова Л.В., Сенько А.В. – Харків: НТУ «ХПІ», 2021. – 112 с
16. Основи програмування на C/C++ в прикладах. Частина 1: навч.-метод. посібник / Соболю М.О., Любченко Н.Ю., Паржин Ю.В., Пугачов Р.В. – Харків : НТУ "ХПІ", 2021. – 113 с
17. Основи програмування Навчальний посібник / Смолій В.М., Лісовиченко О.І. – К.: Видавництво, 2024 – 393 с <https://dglb.nubip.edu.ua/items/980c0b22-f5a3-4bc1-9f04-f8329e0d4bcb>
18. Сьерра К., Бейтс Б. Head First. Java. Легкий для сприйняття довідник / пер. з англ. Харків : Фабула, 2022. 720 с. https://fabulabook.com/info-head-first-java-10000531?srsId=AfmBOopi3crJKwF99mDud4H1AcBhoxqX0q8vg4cifDKK8_7w1ZJBkgcx
19. Фрімен Е., Робсон Е. Head First. Програмування на JavaScript / пер. з англ. Г. Якубовської. Харків : Фабула, 2022. 672 с. https://fabulabook.com/info-head-first-programuvannya-na-javascript-10001968?srsId=AfmBOorOwDoCaHVHi0-kP7gyzkffjsoeYEY3IH_USLcTxVGm3eTuMdYm
20. Шпурик В. В. Навчальний посібник «Основи програмування. Професійний підхід» за освітньо-професійною програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спец. 121 Інженерія програмного забезпечення / В.В. Шпурик, К.М. Оленєва ; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2024. – 106 с <https://ela.kpi.ua/server/api/core/bitstreams/a25c7338-2ff4-4017-8306-8c11c77236f2/content>
21. Юсович-Жуковська В. І., Богут О. М. Web-програмування : Підручник. – Рівне : Волин. обереги, 2023. – 384 с
22. Основи програмування: навчально-методичний посібник [Електронний ресурс]: навч.-метод. посіб. для здобувачів першого (бакалаврського) рівня освітніх програм: «Професійна освіта (цифрові технології)» та «Інженерія програмного забезпечення» Укл.: О.М. Корень, к.т.н., доцент кафедри інформаційних технологій, О.Ф. Кришан, к.е.н., доцент кафедри інформаційних технологій. Кременчук, Придніпровський інститут ПрАТ «ВНЗ «МАУП», 2025. - 282 с.

Додаткова:

1. Кухарчук, В. І. Основи програмування: Навчальний посібник для вищих навчальних закладів. — Львів: Новий Світ-2000, 2018. — 384 с.
2. Eck, D. J. Introduction to Programming Using Java. — 8th ed. — Hobart and William Smith Colleges, 2019. — <http://math.hws.edu/javanotes/>
3. Knuth, D. E. The Art of Computer Programming, Vol. 1: Fundamental Algorithms. — 3rd ed. — Addison-Wesley Professional, 1997. — 672 p.
4. Severance, C. Python for Everybody: Exploring Data in Python 3. — <https://www.py4e.com/>
5. freeCodeCamp.org. Learn to Code — For Free. — <https://www.freecodecamp.org/>
6. Coursera. Programming for Everybody (Getting Started with Python) by University of Michigan. — <https://www.coursera.org/learn/python>
7. Основи мови програмування C++: навчальний посібник. – К.: Логос 2020. – 90 с
8. Веб-ресурс W3Schools.ua. Онлайн-підручники з C++ та веб-технологій (HTML, CSS, JavaScript). Інтерактивні вправи, теоретичні матеріали й вправи для практичного опанування. URL: <https://w3schoolsua.github.io/cpp/index.html>
9. Комлева Н. О. Основи програмування: навчальний посібник / Н. О. Комлева. – Одеса: Наука і техніка, 2020. – 288 с. Лабораторний практикум з програмування мовою C/C++ : навч. посіб. для студ. тех. спец. закл. вищ. освіти I–IV рівн. акредит. / П. А. Пех, С. В. Лавренчук, М. В. Делявський, С. В. Гринюк. – Луцьк : Вежа-Друк, 2020. – 228 с https://lib.lntu.edu.ua/sites/default/files/2021-02/%D0%9F%D0%B5%D1%85_%D0%9B%D0%B0%D0%B2_%D0%94%D0%B5%D0%BB_%D0%93%D1%80%D0%B8%D0%BD_23_12.pdf

10. Професійні стандарти ІТ-фахівців в Україні 2021, де окремо описані компетенції з програмування та розробки ПЗ. URL: <https://mon.gov.ua/static-objects/mon/sites/1/vyshcha/IT-prof-standarty/5-ps-spes-infosystems-13.12.2014.pdf>

11. Кормен, Томас Г. Вступ до алгоритмів : Переклад з англійської третього видання : [укр.] = Introduction to Algorithms : Third Edition : [пер. з англ.] / Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліфорд Стайн, — К. : К. І. С., 2019. — 1288 с
http://library.kpi.kharkov.ua/files/new_postupleniya/vstdoa.pdf

Інформаційні ресурси:

1. Офіційний веб-сайт Python: <https://www.python.org/>
2. Інтерактивні тренажери: <https://www.codewars.com/>, <https://leetcode.com/>
3. Платформа для змагань: <https://www.hackerrank.com/>
4. Спільнота розробників: <https://stackoverflow.com/>
5. Курси та tutorіали: <https://realpython.com/>, <https://www.w3schools.com/python/>