



СИЛАБУС

Базова інформація про дисципліну	
Назва дисципліни	SE023 / Тренінг-курс з основ програмування / Training Course on Programming Fundamentals
Рівень вищої освіти / фахової передвищої освіти	Перший (бакалаврський)
Галузь знань	12 «Інформаційні технології»
Спеціальність	123 «Комп'ютерна інженерія»
Освітня програма	123 «Комп'ютерна інженерія»
Семестр	1
Факультет / відділення	Комп'ютерної інженерії та інформаційних технологій
Курс	1
Анотація курсу	Предмет передбачає ознайомлення та закріплення знань і вмінь з розширеної тематики основ програмування. Він має на меті вивчення базових інструментів управління кодом, вступ до різних парадигм програмування та основи обробки даних. Основною мовою програмування є Python, також задіюються система контролю версій Git, інтерактивне середовище Jupyter Notebook / Google Colab та профільні бібліотеки, зокрема numpy, pandas та ін.
Сторінка курсу в MOODLE	http://78.137.2.119:2929/course/view.php?id=220
Мова викладання	Українська
Лектор курсу	Марченко С.В., викладач I категорії канали комунікації: СДН «Moodle»: повідомлення в чаті E-mail: sv.marchenko1989@gmail.com

Місце дисципліни в освітній програмі	
Освітня програма	https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/123-kompyuterna-inzheneriya.pdf
Перелік загальних компетентностей (ЗК)	<p>Z1. Здатність до абстрактного мислення, аналізу і синтезу.</p> <p>Z2. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>Z3. Здатність застосовувати знання у практичних ситуаціях.</p> <p>Z7. Вміння виявляти, ставити та вирішувати проблеми.</p>
Перелік спеціальних компетентностей (СК)	<p>P2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.</p> <p>P11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.</p> <p>P13. Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.</p> <p>P15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.</p>
Перелік програмних результатів навчання	<p>N8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.</p> <p>N11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.</p> <p>N12. Вміти ефективно працювати як індивідуально, так і у складі команди.</p> <p>N16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.</p>
Опис дисципліни	
Структура навантаження на студента	<p>Загальна кількість годин – 180</p> <p>Кількість кредитів – 6</p> <p>Кількість лекційних годин – 30</p> <p>Кількість практичних занять – 30</p> <p>Кількість годин для самостійної роботи студентів – 120</p> <p>Форма підсумкового контролю – залік</p>

Методи навчання	<p>За подачею навчального матеріалу: методи готових знань, дослідницький метод.</p> <p>З огляду на мету навчання: методи здобуття нових знань, метод формування умінь і навичок, метод застосування знань на практиці, методи закріплення знань, умінь і навичок, методи перевірки і оцінювання знань, умінь і навичок</p>
Зміст дисципліни	
Тема 1. Базові інструменти управління кодом	Базові команди Bash-терміналу. Базові поняття структурної організації програмних проєктів. Різновиди Python-додатків та їх мінімальні реалізації. Менеджери пакетів та організація коду Python-проєкту. Компіляція Python-коду та запуск додатку з консолі. Організація вводу-виводу даних.
Тема 2. Основні сценарії роботи з локальним Git-репозиторієм	Принципи роботи системи контролю версій Git. Створення репозиторію в командному рядку. Вітки та галуження в Git. Візуальні операції з Git. Злиття віток та пов'язані з ним конфлікти. Скасування змін та переміщення по таймлайну розробки.
Тема 3. Основи командної роботи з віддаленим репозиторієм	Базова взаємодія з сервісом GitHub. Оформлення репозиторію. Робочий процес GitHub. Редактори коду та інтеграція з GitHub.
Тема 4. Програми та їх представлення в пам'яті	Представлення інформації в пам'яті комп'ютера. Система типів мови програмування Python. Класи та об'єкти в мові Python. Типізація мов програмування.
Тема 5. Базові засоби IDE для управління кодом	Структура програми мовою Python. Трансляція Python-коду. Байткод та його дизасемблювання. Стандарти форматування коду. Стилiстичне оформлення та документування Python-коду. Стратегії та інструменти налагодження коду. Модульна контрольна робота 1.
Тема 6. Аналіз постановки та хід розв'язування задачі	Аналіз постановки задачі. Алгоритми: поняття та аналіз. Задачі з циклами, умовними та безумовними переходами. Елементарні техніки розробки тестових сценаріїв. Принципи

	модульного тестування коду. Основи роботи з фреймворком unittest.
Тема 7. Функції та елементи функціонального програмування	Організація Python-коду за допомогою функцій. Виконання програми в функціональному стилі. Функції, ітератори, генератори. Функції вищого порядку.
Тема 8. Робота з датами та часом	Проблематика роботи з датами й часом. Python-бібліотеки для роботи з датами й часом. Програмування дат і часу. Арифметика дат. Робота з часовими поясами. Робота з календарем.
Тема 9. Робота з текстом	Функції для роботи з текстом. Регулярні вирази.
Тема 10. Вбудовані структури даних	Кортежі, списки, множини, словники, стеки, черги. Бібліотека numpy: вектори, матриці, масиви. Елементи роботи з зображеннями.
Тема 11. Робота з файлами та елементарна обробка даних	Основи файлового вводу-виводу в Python. Серіалізація об'єктів у мові Python. Консервування об'єктів. Робота з серіалізаційним представленням json. Робота з серіалізаційним представленням csv. Базове влаштування та організація доступу до даних. Завантаження та нормалізація даних. Базові перетворення даних у pandas. Фільтрування даних. Групування та агрегація даних. Модульна контрольна робота 2.
Політика дисципліни	
Політика відвідування	Регулярне відвідування всіх видів занять, своєчасність виконання самостійної роботи. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання зорганізується в он-лайн формі за погодженням із керівником курсу.
Політика щодо дедлайнів та перескладання	Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.
Академічна доброчесність	У випадку недотримання політики академічної доброчесності (плагиат, самоплагиат, фабрикація, фальсифікація, списування, обман,

хабарництво) передбачено повторне проходження оцінювання.

Система оцінювання

Поточний контроль здійснюється протягом семестру під час проведення практичних, семінарських та інших видів занять і оцінюється сумою набраних балів (максимальна сума – 100 балів; мінімальна сума, що дозволяє студенту отримати атестацію з предмету – 60 балів); підсумковий/ семестровий контроль, проводиться у формі заліку або іспиту, відповідно до графіку навчального процесу.

Підсумкова оцінка за умови заліку виставляється як загальна сума балів, набраних за результатами поточного контролю. Підсумкова оцінка за умови іспиту виставляється як загальна сума балів набраних за результатами поточного (70%) та підсумкового контролю.

Накопичування рейтингових балів з навчальної дисципліни

Види навчальної роботи											Мах кількість балів
Аудиторна											
Практичні завдання (11 тем)											62
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	
5	6	5	6	4	8	6	4	4	7	7	
Тестування											8
Індивідуальна											
Захист практичних завдань до самостійної роботи (11 тем)											30
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	
-	-	6	-	4	-	-	4	-	8	8	
РАЗОМ											100

Шкала оцінювання

ECTS	Бали	Зміст
A	90-100	Бездоганна підготовка в широкому контексті
B	80-89	Повні знання, міцні вміння
C	70-79	Хороші знання та вміння
D	65-69	Задовільні знання, стереотипні вміння
E	60-64	Виконання мінімальних вимог діяльності в стандартних умовах
FX	35-59	Слабкі знання, відсутність умінь
F	1-34	Необхідний повторний курс

Список рекомендованих джерел

Основна література

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Чернігів: ФОП Баликіна С.М., 2020. 180 с.
2. Руденко В.Д., Жугастров О.О. Основи алгоритмізації і програмування мовою Python. Київ: Ранок, 2019. 192 с.
3. Chacon S., Straub B. Pro Git (переклад українською, версія 2.1.4). NY: Apress, 2021. 548с.
4. Lubanovic B. Introducing Python: Modern Computing in Simple Packages 2nd Edition. Sebastopol: O'Reilly Media, 2019. 606р.
5. Matthes E. Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 2nd Edition. San Francisco: No Scratch Press, 2019. 544с.
6. Steinhart J.E. The Secret Life Of Programs: Understand Computers — Craft Better Code. San Francisco: No Starch Press, 2019. 458с.
7. Lott S., Phillips D. Python Object-oriented Programming. Fourth Edition. Birmingham: PACKT Publishing, 2021. 694 с.
8. Lott S. Functional Python Programming. Create succinct and expressive implementations with functional programming in Python. Birmingham: PACKT Publishing, 2018. 408 с.
9. Трофименко О. Г., Прокоп Ю. В., Задерейко О. В. Алгоритмізація та програмування : навчально-методичний посібник. Одеса : Фенікс, 2020. 310 с.
10. Stepanek H. Thinking in Pandas. How to Use the Python Data Analysis Library the Right Way. NY: Apress, 2020. 186р.

Допоміжна література

1. Barrett D. J. Efficient Linux at the Command Line. Sebastopol: O'Reilly Media, 2022. 232р.
2. Ramalho L. Fluent Python. Clear, Concise, and Effective Programming, 2nd Edition. Sebastopol: O'Reilly Media, 2022. 1012р.
3. Phillips D. Python 3 Object-oriented Programming. Third Edition. Birmingham: PACKT Publishing, 2018. 466р.
4. Steinhart J. E. The Secret Life Of Programs: Understand Computers — Craft Better Code / San Francisco: No Starch Press, 2019. – 458р.
5. Tsitoara M. Beginning Git and GitHub. A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer. New York: Apress, 2020. 290р.
6. Harrison M., Petrou Th. Pandas 1.x Cookbook. Practical recipes for scientific computing, time series analysis, and exploratory data analysis using Python, Second Edition. Birmingham: PACKT Publishing, 2020. 606р.

Інформаційні ресурси

1. Програмування для всіх (початок роботи з Python) [Електронний ресурс]. URL: <https://www.coursera.org/learn/python> (дата звернення: 29.08.2023р.).
2. Pythonic way. Don't learn to code. Code to learn! [Електронний ресурс]. URL: <http://pythonicway.com/> (дата звернення: 29.08.2023р.).
3. Python Tutorials [Електронний ресурс]. URL: <https://www.youtube.com/playlist?list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU> (дата звернення: 29.08.2023р.).
4. Уроки Python для початківців / Програмування на Пітон з нуля [Електронний ресурс]. URL: <https://www.youtube.com/playlist?list=PL7vq4D0vOpQb-nious5dTzUF9GEMp6jiL> (дата звернення: 29.08.2023р.).
5. Вивчай Git Branching [Електронний ресурс]. URL: <https://learngitbranching.js.org/?locale=uk> (дата звернення: 29.08.2023р.).
6. Version Control with Git. URL: <https://www.coursera.org/learn/version-control-with-git> (дата звернення: 29.08.2023р.).
7. Спеціалізація Python for Everybody. URL: <https://www.coursera.org/specializations/python> (дата звернення: 29.08.2023р.).