ADVANCED PYTHON PROGRAMMING FOR DATA SCIENCE

ENCT 325

Lecture	**: 3**	**Year : III**
Tutorial	**: 2**	**Part : I**
Practical	**: 1**	

Course Objectives:

The objective of this course is to develop advanced proficiency in Python programming for data science applications. It focuses on efficient coding practices, sophisticated data manipulation, statistical analysis, and data visualization using modern Python libraries. Students will also learn fundamental data engineering and pipeline design concepts to automate and scale real-world data workflows.

1 Advanced Python Concepts and Best Practices (7 hours)

1.1 Review of Python essentials and coding conventions
1.2 Advanced data structures: Collections, iterators, generators, and decorators
1.3 Functions and lambda expressions
1.4 Object-Oriented Programming for data science applications
1.5 Exception handling, debugging, and logging
1.6 Working with modules and packages

2 Data Sources and APIs (7 hours)

2.1 Reading and writing structured/unstructured data (CSV, JSON, Excel, text)
2.2 Database access with relational database and non-relational database
2.3 Accessing and processing data from APIs (REST, SOAP)
2.4 Web scraping using requests and BeautifulSoup
2.5 Handling large datasets with chunking and lazy evaluation

3 Advanced Data Wrangling and Transformation (9 hours)

3.1 Advanced Pandas operations: Merging, joining, reshaping, pivoting
3.2 Handling missing, categorical, and time-series data
3.3 Feature transformation, scaling, and encoding
3.4 Memory optimization and efficient data processing
3.5 Building a reusable data-cleaning pipeline
3.6 Introduction to data pipeline components (Ingestion, transformation, storage)

4 Applied Statistics and Exploratory Analysis (7 hours)

4.1 Statistical measures: Correlation, covariance, skewness, kurtosis
4.2 Probability review, sampling, and hypothesis testing
4.3 Regression and trend analysis using stats models
4.4 Exploratory data analysis (EDA) using descriptive and inferential methods
4.5 Automation of EDA workflows using Python

5 Data Visualization and Storytelling (7 hours)

5.1 Principles of effective visualization and dashboard design
5.2 Visualization with Matplotlib: Line, bar, histogram, scatter, subplots
5.3 Seaborn for statistical visualization: Box plot, pair plot, heat map
5.4 Interactive visualization using Plotly
5.5 Visualization driven insight generation
5.6 Case study: End-to-end visualization and reporting project

6 Data Engineering and Automation (8 hours)

6.1 Overview of data engineering in applied data science
6.2 Designing and implementing ETL pipelines
6.3 Automating workflows with schedulers (CRON, schedule)
6.4 Logging, monitoring, and error handling in pipelines
6.5 Data storage and retrieval strategies for pipelines
6.6 Automated report generation (Excel, HTML, PDF)
6.7 Case study: End-to-end automated analytics pipeline

Tutorial (30 hours)

1. Python refresher and best practices: Review of Python syntax, indentation, and PEP8 coding conventions through short exercises
2. Iterators, generators, and decorators: Writing small programs using generators for data streaming and decorators for function modification
3. Object-oriented programming (OOP): Designing simple class-based programs and demonstrating inheritance and encapsulation in python
4. Data access and integration: Reading data from CSV, Excel, JSON, and APIs; Discussion on best practices for data ingestion
5. Database and SQL interaction: practice using SQLite and SQLAlchemy to query and manipulate structured datasets
6. Web scraping practice: Extracting tabular data using requests and BeautifulSoup; Handling exceptions and encoding issues
7. Advanced Pandas operations: Hands-on merging, reshaping, pivoting, and group-by operations for complex data manipulation
8. Data cleaning and transformation: Exercises on handling missing values, encoding categorical data, and normalization techniques
9. Statistical computation and EDA: Performing descriptive analysis, correlation, and hypothesis testing using Python libraries

10. Data visualization practice: Creating comparative plots using matpLotlib, Seaborn, and Plotly; Customizing themes and layouts
11. Pipeline and automation concepts: Designing pseudocode and flow diagrams for ETL data pipelines and discussing error handling strategies
12. Mini case study discussion: Guided review of a small end-to-end applied data pipeline from ingestion to visualization and reporting

Practical **(15 hours)**

1. Setting up Python environment for applied data workflows and writing modular programs using OOP and functions
2. Collecting data via APIs and web scraping
3. Building advanced data cleaning and transformation pipelines using Pandas
4. Conducting exploratory data analysis and statistical summaries
5. Developing interactive visualizations using Plotly, matplotlib and Seaborn
6. Automating ETL tasks and data refresh using Python schedulers
7. Generating summary dashboards and automated analytical reports
8. Mini Project: Build a complete applied data pipeline from ingestion to visualization and reporting on a real-world dataset

Final Exam

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

Chapter	**Hours**	**Marks distribution***
1	7	9
2	7	9
3	9	12
4	7	10
5	7	9
6	8	11
Total	**45**	**60**

* There may be minor deviation in marks distribution.

References

1. McKinney, W. (2022). Python for data analysis. O’Reilly Media.
2. VanderPlas, J. (2016). Python data science handbook. O’Reilly Media.
3. Beazley, D. (2021). Python cookbook. O’Reilly Media.
4. Grus, J. (2022). Data science from scratch. O’Reilly Media.