**Advanced Data Analytics with Python and SQL**

**Introduction to Python**

**Objective:** Gain a thorough understanding of Python's history, features, and environment setup.

**Topics:**

1. **Overview of Data Analytics**
   * Introduction to Data Analytics
   * Importance of Python in Data Analytics
   * Real-world Applications
2. **Python Programming History & Features**
   * History of Python
   * Key Features of Python
3. **Setting Up a Python Environment**
   * Installing Anaconda
   * Introduction to Jupyter Notebooks
   * Setting Up Visual Studio Code for Python
   * Introduction to Pycharm
4. **Python Syntax Overview**
   * Basic Syntax
   * Indentation
   * Comments and Docstrings

**Basic Python**

**Objective**: Master the fundamental elements of Python programming.

**Topics:**

1. **Identifiers and Variables**
   * Naming Conventions
   * Assigning Values
   * Dynamic Typing
2. **Keywords**
   * List of Python Keywords
   * Reserved Words
3. **Operators**
   * **Arithmetic Operators:**
     + Addition, Subtraction, Multiplication, Division, Modulus
   * **Comparison Operators:**
     + Equal to, Not equal to, Greater than, Less than
   * **Logical Operators:**
     + and, or, not
   * **Bitwise Operators:**
     + AND, OR, XOR, NOT, Shift Operators
   * **Assignment Operators:**
     + =, +=, -=, \*=, /=, %=, \*\*=, //=
   * **Identity Operators:**
     + is, is not
   * **Membership Operators:**
     + in, not in
4. **Data Types**
   * **Primitive Data Types:**
     + Integer
     + Float
     + String
     + Boolean
   * **Non-Primitive Data Types:**
     + Array
     + List
     + Tuple
     + Dictionary
     + Sets
5. **Comprehension in Python**
   * List Comprehensions
   * Dictionary Comprehensions
   * Set Comprehensions
   * Nested Comprehensions

**Control Flow**

**Objective:** Understand and implement control flow mechanisms.

**Topics:**

1. **Conditional Statements**
   * If Statement
   * If-else Statement
   * If-elif-else Statement
   * Nested if Statements
2. **Loops**
   * While Loop
   * For Loop
   * Break Statement
   * Continue Statement
   * Pass Statement

**Functions and Modules**

**Objective:** Define and use functions and modules to create modular code.

**Topics:**

1. **User Defined Functions**
   * Defining Functions
   * Function Arguments
   * Return Statement
2. **Built-in Functions**
   * Common Built-in Functions
   * Using Built-in Functions
3. **Lambda Functions**
   * Anonymous Functions
   * Syntax and Usage
4. **Map, Filter, Reduce**
   * **Map:**
     + Applying a function to all items in an input list
   * **Filter:**
     + Constructing a list from those elements of the input list for which a function returns true
   * **Reduce:**
     + Applying a rolling computation to sequential pairs of values in a list

**File Handling**

**Objective:** Perform file operations in Python for reading and writing data.

**Topics:**

1. **File Operations**
   * Overview of file handling in Python
   * Importance of file handling in programming
   * Files like csv, excel, text, pdf, json.
2. **Opening Files**
   * **Using the open() function**
     + Syntax: open(filename, mode)
     + Different modes for opening files( r, w, a, x)
3. **Creating Files**
   * Creating a new file using 'w', 'a', or 'x' mode
4. **Reading Files**
   * Reading the entire content using read()
   * Reading line by line using readline()
   * Reading all lines into a list using readlines()
5. **Writing to Files**
   * Writing a string to a file using write()
   * Writing multiple lines using writelines()
   * Importance of newline characters (\n) in text files
6. **Deleting Files**
   * Using the os module to delete files

**Exception Handling**

**Objective:** Handle exceptions and errors gracefully in Python.

**Topics:**

1. **Types of Errors**
   * Syntax Errors
   * Runtime Errors
   * Logical Errors
2. **Exception Handling**
   * try … except Block
   * try … except … finally Block
   * try … except … else Block
   * Handling Multiple Exceptions
   * Raising Exceptions

**Regular Expressions**

**Objective:** Use regular expressions for pattern matching in strings.

**Topics:**

1. **Python re Module**
   * Functions in re Module
   * Compiling Regular Expressions
2. **Methods with Regex Usage**
   * match()
   * search()
   * findall()
   * sub()
   * split()

**Object-Oriented Programming (OOP) in Python**

**Objective:** Master the core concepts of OOP in Python for designing modular code.

**Topics:**

1. **Class and Object**
   * Defining Classes
   * Creating Objects
   * Class Attributes and Methods
2. **Polymorphism**
3. **Encapsulation**
4. **Inheritance**

**Statistics with Python**

**Objective:** Understand basic statistical concepts and perform statistical analysis using Python.

**Topics:**

1. **Introduction to Statistics**
   * Importance of Statistics in Data Analysis
   * Types of Statistics: Descriptive and Inferential
2. **Descriptive Statistics**
   * Measures of Central Tendency: Mean, Median, Mode
   * Measures of Dispersion: Range, Variance, Standard Deviation
   * Skewness and Kurtosis
3. **Probability**
   * Basic Probability Concepts
   * Probability Distributions: Normal, Standard Normal Distribution
4. **Correlation and Regression**
   * Correlation Coefficient
   * Coefficient of Determination
   * Simple Linear Regression
5. **Hands-on:**
   * Implementing Statistical Measures in Python
   * Using Libraries like NumPy, SciPy, and StatsModels

**NumPy**

**Objective:** Introduction to NumPy for numerical operations.

**Topics:**

1. **Difference Between Numpy and List**
2. **Introduction to NumPy**
   * NumPy Array
   * numpy.random Modules
   * Array Operations
   * Vector Operation
   * Statistics Function
   * Array Indexing
   * Array Manipulation
   * Array Broadcasting
3. **Hands-on:**
   * Practice with NumPy Arrays
   * Mathematical Operations with NumPy

**Data Pre-processing with Pandas**

**Objective:** Manipulate and preprocess data using Pandas.

**Topics:**

* **Introduction to Pandas Library**
  + Series and DataFrame
  + Data Structures in Pandas
* **Working with Series and Data Frames**
  + Creating Series and DataFrames
  + Basic Operations on Series and DataFrames
* **Indexing and Selecting Data**
  + Selecting Rows and Columns
  + Filtering Data
* **Data Cleaning and Preprocessing**
  + Dealing with Duplicate Data
  + Handling Outliers
  + Feature Scaling and Normalization
  + Encoding Categorical Variables
* **Pandas Methods**
  + **Creating DataFrames**
    - From a dictionary: pd.DataFrame(data)
    - From a CSV file: pd.read\_csv('filename.csv')
    - From a EXCEL file: pd.read\_excel('filename.xlsx')
    - From a SQLfile: pd.read\_sql('filename.sql')
  + **Viewing Data**
    - First few rows: df.head()
    - Last few rows: df.tail()
    - Retrieve column names: df.columns
    - Summary info: df.info()
    - shape of dataframe: df.shape
    - Basic statistics: df.describe()
    - Get unique values in a column: df['column'].unique()
    - Count unique values in a column: df['column'].nunique()
    - Get n smallest values in a column: df['column'].nsmallest()
    - Get n largest values in a column: df['column'].nlargest()
  + **Selecting Data**
    - Single column: df['column\_name']
    - Multiple columns: df[['col1', 'col2']]
    - Rows by label: df.loc[label]
    - Rows by position: df.iloc[position]
  + **Filtering Data**
    - Based on condition: df[df['column'] > value]
    - Multiple conditions: df[(df['col1'] > val1) & (df['col2'] == val2)]
    - Filter rows using query: df.query()
    - Rename columns: df.rename(columns={})
  + **Adding/Modifying Columns**
    - Add new column: df['new\_col'] = values
    - Modify existing column: df['col'] = df['col'] + value
  + **Removing Data**
    - Remove columns: df.drop('col\_name', axis=1, inplace=True)
    - Remove rows: df.drop(index, axis=0, inplace=True)
  + **Handling Missing Data**
    - Detect missing values: df.isnull()
    - Drop rows with missing values: df.dropna()
    - Fill missing values: df.fillna(value)
    - Detect duplicate rows: df.duplicated()
    - Drop duplicate rows: df.drop\_duplicates()
  + **Aggregation and Grouping**
    - Group by a column or columns: df.groupby()['column']
    - Aggregate functions: grouped.mean(), grouped.sum(), grouped.size()
    - Count occurrences of each unique value: df['column'].value\_counts()
    - Sort values in a column: df['column'].sort\_values()
    - Sort index labels: df.sort\_index()
  + **String Methods**
    - Convert to lowercase: df['col'].str.lower()
    - Convert to uppercase: df['col'].str.upper()
    - Strip whitespace: df['col'].str.strip()
    - Replace substring: df['col'].str.replace('old', 'new')
    - Contains substring: df['col'].str.contains('substring')
    - Extract substring: df['col'].str.extract('regex')
    - Split strings: df['col'].str.split('delimiter')
    - Concatenate strings: df['col1'] + ' ' + df['col2']
    - Starts with substring: df['col'].str.startswith('prefix')
    - Ends with substring: df['col'].str.endswith('suffix')
    - Find position of substring: df['col'].str.find('substring')
    - Count occurrences of substring: df['col'].str.count('substring')
    - Replace using regex: df['col'].str.replace(r'regex', 'new', regex=True)
    - Find all matches of a regex: df['col'].str.findall('regex')
  + **Merging and Joining**
    - Merge DataFrames: pd.merge(df1, df2, on='key')
    - Concatenate DataFrames: pd.concat([df1, df2], axis=0)
  + **Date and Time Handling**
    - Convert to datetime: pd.to\_datetime(df['date\_col'])
    - Extract date parts: df['Year'] = df['date\_col'].dt.year
  + **Pivot Tables**
    - Create pivot table: df.pivot\_table(values='val\_col', index='idx\_col', columns='col\_to\_pivot')
  + **Exporting Data**
    - To CSV file: df.to\_csv('filename.csv', index=False)
    - To Excel file: df.to\_excel('filename.xlsx', index=False)

**Data Visualization with Matplotlib and Seaborn**

**Objective:** Create visualizations using Matplotlib and Seaborn.

**Topics:**

1. **Introduction to Data Visualization**
   * Importance of Data Visualization
   * Types of Data Visualization
2. **Matplotlib for Basic Plotting**
   * Line Plot
   * Bar Plot
   * Histogram
   * Scatter Plot
   * Pie Chart
   * Box and whiskers Plot
3. **Seaborn for Statistical Data Visualization**
   * Line Plot
   * Barplot
   * Boxplot
   * Heatmap
   * Pairplot
   * Countplot
   * regplot
   * Scatterplot
   * Hueplot
   * Violin plot
   * Swarmplot
   * Stipplot
4. **Customizing Plots and Charts**
   * Choosing axis
   * Adding grids
   * Customizing axis values
   * Adding Titles and Labels
   * Customizing Colors and Styles
   * Adding Legends

**Exploratory Data Analysis (EDA)**

**Objective:** Understand and perform exploratory data analysis (EDA) to summarize the main characteristics of a dataset and uncover patterns, spot anomalies, test hypotheses, and check assumptions using visualizations and summary statistics.

**Topics:**

* **Introduction to Exploratory Data Analysis (EDA)**
  + Introduction to EDA
  + Tools and Libraries for EDA
  + Loading Data
  + **Data Cleaning and Preparation**
    - Identifying and handling Missing Data
    - Identifying and handling Duplicates
    - Identifying and handling Outliers
* **Feature Engineering**
  + One hot encoding
  + Label encoding
  + Range Categorization
* **Univariate Analysis**
  + Summary Statistics
  + Visualizations for Univariate Analysis
  + Distribution Analysis
* **Bivariate Analysis**
  + Summary Statistics for Bivariate Analysis
  + Visualizations for Bivariate Analysis
  + Categorical vs. Numerical Analysis
* **Multivariate Analysis**
  + Summary Statistics for Multivariate Analysis
  + Visualizations for Multivariate Analysis
  + Adding Legends
* **Exploratory Data Analysis (EDA) Practice**
  + Case Study
  + Reporting and Presenting EDA Findings
* **Hands-on**
  + Conducting a complete EDA on a given dataset
  + Creating and presenting an EDA report
* **Mentored EDA Projects Hands-On**
  + Analyzing Diwali Sales Trends
  + IPL Match Performance Analysis
  + Zomato Restaurant Data Exploration
  + Sales Insights from Euromart Data
  + Car Manufacturing and Pricing Analysis
  + Titanic Survival Data Analysis
  + Exploring IMDB Movie Ratings and Trends
  + Global Store Sales Performance Analysis

**Final Project**

**Objective:** Apply all learned concepts to a real-world dataset and complete a comprehensive data analysis project.

**Tasks:**

1. **Creating a Complete Data Analysis Project**
   * Data Collection
   * Data Cleaning
   * Data Analysis
   * Data Visualization
   * EDA
2. **Presenting Findings and Insights**
   * Summarizing Key Insights
   * Presenting Visualizations and Analysis
3. **Documentation**
   * Documenting the Project
   * Writing a Detailed Report

**Extract Transform Load (ETL): Integrating MySQL with Python**

***This module will be taught only after detailed MySQL module is completed***

**Objective:** Interact with MySQL databases using Python.

**Topics:**

1. **SQL Statements & Operations**
   * Create, Read, Update, Delete Operations
   * SQL Syntax and Queries
2. **Python-SQL Connector Package Installation**
   * Installing MySQL Connector
   * Setting Up Database Connection
3. **Python with CRUD Operations**
   * Creating Records
   * Reading Records
   * Updating Records
   * Deleting Records