Course Title:

Python for Data Science

Course Description:

This course provides a comprehensive introduction to Python programming, focusing on its applications in data science. Students will learn essential Python concepts, including data structures, conditional statements, loops, functions, and object-oriented programming. The course also covers data science libraries like NumPy, Pandas, Seaborn, and Matplotlib, with hands-on labs where students will work on real-world problems using Jupyter Notebooks.

Course Level:

Beginner to Intermediate (No prior knowledge of Python is required, but some basic programming familiarity is helpful)

Duration:

5 Weeks (Two 1.5-hour classes per week)

Course Objectives:

By the end of this course, learners will:

- Understand the basic and intermediate-level Python concepts.
- Learn how to set up and use Jupyter Notebooks for Python programming.
- Gain hands-on experience with essential Python libraries for data science.
- Write Python programs using variables, data structures, loops, functions, and objects.
- Use popular data science libraries (NumPy, Pandas, Seaborn, Matplotlib) for data manipulation and visualization.

Course Schedule:

Week 1:

Lecture 1: Introduction to Python and Setting Up Jupyter Notebooks

- Introduction to Python
- Installing Anaconda and setting up Jupyter Notebooks
- Understanding Variables and Data Types in Python
- Lab Work: Hands-on setup of Jupyter Notebooks and writing your first Python code.

Lecture 2: Lists, Tuples, and Sets in Python

- What are Lists, Tuples, and Sets?
- Working with Lists, Tuples, and Sets (creation, access, and modification)
- Lab Work: Writing code for data structures and performing basic operations.

Week 2:

Lecture 3: Conditional Statements and Loops

- Introduction to If-Else Statements
- For and While Loops in Python
- Nested conditions and loops
- Lab Work: Writing code with conditional statements and loops for basic problem solving.

Lecture 4: Functions, Classes, Objects, and Packages in Python

- Defining and using Functions in Python
- Introduction to Object-Oriented Programming (OOP): Classes and Objects
- Packages and libraries in Python
- Lab Work: Writing and using functions, creating classes, and working with Python packages.

Week 3:

Lecture 5: Introduction to NumPy

- Overview of NumPy and its importance in Data Science
- Creating arrays and performing mathematical operations using NumPy
- Lab Work: Working with NumPy arrays, performing array operations.

Lecture 6: Working with NumPy Arrays

- Advanced NumPy operations
- Indexing, slicing, and reshaping arrays
- Lab Work: Practical coding exercises on manipulating NumPy arrays.

Week 4:

Lecture 7: Introduction to Pandas for Data Manipulation

- What is Pandas? Understanding DataFrames and Series
- Reading and manipulating data using Pandas
- Lab Work: Importing datasets and performing basic data operations using Pandas.

Lecture 8: Advanced Data Manipulation with Pandas

- Cleaning, filtering, and grouping data with Pandas
- Merging and concatenating datasets
- Lab Work: Writing code to handle real-world datasets and performing advanced Pandas operations.

Week 5:

Lecture 9: Introduction to Data Visualization with Seaborn and Matplotlib

- What is Data Visualization? Importance in Data Science
- Introduction to Matplotlib and Seaborn
- Plotting basic charts like line plots, bar charts, and histograms
- Lab Work: Creating visualizations using Seaborn and Matplotlib.

Lecture 10: Advanced Data Visualization

- Creating advanced plots: Heatmaps, pair plots, and scatter plots
- Customizing visualizations for better insights
- Lab Work: Writing code to generate complex data visualizations for real datasets.

Resources Provided by the Instructor:

Throughout the course, students will be provided with Jupyter Notebooks containing Python coding problems and exercises tailored to each topic covered in class. These notebooks will serve as valuable practice material, allowing students to reinforce their understanding of Python programming, data manipulation, and visualization techniques. Additionally, the notebooks will include step-by-step instructions, sample code, and guided solutions to ensure that students can progress confidently through the course content. These resources will be made available after each lecture to support both in-class learning and independent study.