Trainer Name: Er. Manish Singh [M.Tech(CSE), B.Tech(CSE)]
Trainer Experience: 12+ Years of IT Industry

Course Overview:

Python Version: 3.14
Training Mode: Hands-on Offline classroom program
Training Duration: Regular 45 Days (2 Hours Per Day)
Training Fees: ₹3500
Training Deliverables:

- Daily Practical Assignment
- Per Day Lab Sessions
- Interview Oriented Programs
- One Complete Real-Time Working Project
- Course Completion Certificate from Naomika Computer Consultancy

About Python Language:

Python is a general-purpose programming language, which means it can be used to build almost any type of software or application. Python is widely used in web development through frameworks like **Django** and **Flask**, which help developers create secure, scalable, and feature-rich web applications quickly. In recent years, Python has become the heart of modern technologies such as **Artificial Intelligence, Machine Learning, and Data Science**. Libraries like NumPy, Pandas, TensorFlow, PyTorch, and Scikit-Learn allow developers to analyze large datasets, build predictive models, and create intelligent systems with minimal effort.

Objective of Python Language Course:

The main objective of choosing Python language over other programming courses is to provide learners with a powerful yet simple pathway into the world of coding and modern technology. Unlike many programming languages that have complex syntax and steep learning curves, Python focuses on readability and ease of understanding. This allows beginners to grasp programming concepts faster while still giving advanced learners the ability to build high-level applications. Another key objective is to prepare students for **high-demand career fields** such as data science, artificial intelligence, machine learning, automation, cybersecurity, web development, and cloud computing.

Why Do People Use Python Today?

The major reason people choose Python is its **versatility**. The same language can be used for web development, data science, machine learning, artificial intelligence, automation, cybersecurity, software testing, IoT, and more. This makes Python a universal choice across industries. Python also provides thousands of **libraries and frameworks**, such as NumPy, Pandas, Django, Flask, TensorFlow, and PyTorch. These tools save time and help professionals build advanced applications faster.

Course Contents:

- Lists
  - List Object Basics
  - Lists in Action
    - ✓ Basic List Operations
    - ✓ Indexing and Slicing
    - ✓ Changing Lists in Place
    - ✓ List Methods
    - ✓ Iteration, Comprehensions, and Unpacking
- Dictionaries
  - Dictionary Object Basics
  - Dictionaries in Action
    - ✓ Basic Dictionary Operations
    - ✓ Changing Lists in Place
    - ✓ Dictionary Methods
    - ✓ Key Insertion Ordering, Union Operator, and Comprehensions
- Tuples and Files
  - Tuple Object Basics
  - Tuples in Action
    - ✓ Basic Tuple Operations
  - Why Tuples?
  - Files
    - ✓ Opening and Closing Files
    - ✓ Reading from Files
    - ✓ Writing to Files
    - ✓ File Position Methods
    - ✓ Other File Management

Module 3: Statements and Syntax

- Introducing Python Statements
  - Python’s Statement
  - What Python adds and Removes
  - Why Indentation?
  - Assignments
    - ✓ Assignment Syntax
    - ✓ Basic Assignments
    - ✓ Sequence Assignments
    - ✓ Extended Unpacking Assignments
    - ✓ Multiple Target Assignments
    - ✓ Augmented Assignments
    - ✓ Named Assignment Expressions
    - ✓ Variable Naming Rules
  - Expression Statements and In-Place Changes
  - Print Operations
- if and match Selections
  - if Statements
    - ✓ General Format and Basic Examples
    - ✓ Multiple Choice Selections
  - match Statements
    - ✓ Basic and Advanced match usage
  - The if/else Ternary Expression
  - Truth Values
- while and for loops
  - while Loops
    - ✓ General Format and Basic Examples
  - for Loops
    - ✓ General Format and Basic Examples

- break, continue, pass and Loop else
- Iterations, Comprehensions and Python Documentation

Module 4: Functions and Generators

- Function Basics
  - Why Use Functions?
  - General Function Concept
  - Function Definition and Call with Examples
- Scopes
  - ✓ Scopes Basics
  - ✓ Name Resolutions: The LEGB Rule
  - ✓ Scopes Examples
  - ✓ The Global Statement
  - ✓ Nested Functions and Scopes
  - ✓ The nonlocal Statement
  - ✓ Scopes and Argument defaults
- Arguments
  - ✓ Argument-Passing Basics
  - ✓ Special Argument-Matching Modes
  - ✓ Argument Ordering
- Function Odds and Ends
  - ✓ Recursive Functions
  - ✓ Function Tools
  - ✓ Anonymous Functions: lambda
  - ✓ Functional Programming Tools
- Comprehensions and Generations
  - ✓ Formal Comprehension Syntax
  - ✓ Generator Functions and Expressions
  - ✓ Asynchronous Functions

Module 5: Modules and Packages

- Modules
  - What is Module?
  - Why Use Modules?
  - Python Program Architecture
    - ✓ How to Structure a Program?
    - ✓ Imports and Attributes
  - How Imports Work
  - Module Search Path
  - Creating Modules
  - Using Modules
  - Modules Namespaces
  - Reloading Modules
- Packages
  - What is Package?
  - Using Packages
  - Creating Packages
  - Why Packages?
  - Namespace Packages

Module 6: Classes and OOPs

- OOPs
  - What is OOPs?
  - Principles of OOPs
  - Class and Object

- Why Use Classes?
- Class Coding
    - Creating Class
    - Instantiation
    - Methods
    - Constructors
    - Inheritance
    - Polymorphism
    - Namespaces

Module 7: Exceptions Handling

- Exceptions
    - Exception Basics
    - Why Use Exceptions?
    - The try Statement
    - The raise Statement
    - The assert Statement
    - The with Statement
    - Exception Objects
    - Exception Classes
    - Built-in Exception Classes

Module 8: Advanced Topics

- Date and Time Modules
    - How to use Date & Date Time class?
    - How to use Time Delta object?
    - Formatting Date and Time
    - Calendar module
    - Text calendar
    - HTML calendar
- OS Modules
    - Shell script commands
    - Various OS operations in Python
    - Python file system shell methods
    - Creating files and directories
    - Removing files and directories
    - Shutdown and Restart system
    - Renaming files and directories
    - Executing system commands
- Unicode and Byte Strings
    - Unicode Foundation
    - Python String Tools
    - Using Text Strings
    - Using Byte Strings
    - Using Text and Binary Files
- Decorators
    - What is Decorator?
    - Basics of Decorator
    - Coding Function Decorators
    - Coding Class Decorators