HDTC Training Center

Provide

Technical proposal of the Training program

Python Programming – Beginner to Intermediate Level

Course Overview:

This comprehensive training course introduces participants to the fundamentals of Python programming and guides them through intermediate concepts, preparing them to write clean, efficient, and maintainable code. The course blends theory with practical applications, using real-world examples, and is ideal for participants looking to build programming skills for software development, data analysis, automation, or web development.

General Objective:

To equip participants with the foundational and intermediate skills in Python programming necessary to build applications, automate tasks, and work with data in a professional environment.

Program Objectives:

By the end of this program, participants will be able to:

- Understand Python syntax, variables, data types, and control structures
- Write functions, use built-in modules, and create reusable code
- Work with files, handle exceptions, and debug Python programs

- Use object-oriented programming (OOP) principles in Python
- Apply Python for basic automation and scripting
- Manipulate and analyze data using standard libraries like pandas and matplotlib (optional)
- Build simple CLI or script-based applications.

Program Outlines:

Day 1: Python Basics

- Introduction to Python and its ecosystem
- Installing Python and setting up development environments (IDLE, VS Code, Jupyter)
- Data types: strings, numbers, booleans
- Variables and basic operators
- Input/output, comments, and code structure
- Control flow: if, else, elif
- Practice exercises: calculator, grade checker

Day 2: Loops, Collections & Functions

- for and while loops
- Working with lists, tuples, sets, and dictionaries
- List comprehensions
- Defining and calling functions
- Function parameters, return values, and scope
- Practice: menu system, to-do list, simple data organizer

Day 3: Files, Exceptions, and Modules

- Reading and writing files (text and CSV)
- Try/except for error handling
- Importing built-in modules (math, random, datetime, etc.)
- Creating your own Python modules

- Practice: log parser, CSV data cleaner

Day 4: Object-Oriented Programming (OOP)

- Introduction to classes and objects
- Constructors (__init__) and instance attributes
- Methods and encapsulation
- Inheritance and polymorphism
- Practice: user management system, class-based projects

Day 5: Scripting & Intermediate Use Cases

- Command-line arguments and simple scripting
- Introduction to working with pandas for data manipulation (optional)
- Visualization with matplotlib (optional)
- Automating tasks (file operations, email sending, data cleaning)
- Final mini project: combine concepts in a working Python script
- Wrap-up and Q&A

Target Audience:

- Beginners with no or little coding background
- Junior developers seeking to learn Python
- Data analysts and business professionals wanting to automate tasks
- Students in technical or scientific fields
- Anyone preparing for advanced Python/data science/web development training

Training methods:

- Technology-Based Learning.
- Simulation in Training.

- On-the-job guidance.
- Trainer-Led Training.
- Work Teams and Roles.
- Films and Videos.
- Case Studies and Workshops.