Python Programming for Clinical / Medical Domain

Duration: 5 days
Prerequisite: Knowledge of programming

Day 1 – Python Refresher

Topics:

- Python refresher:
  - Data types (str, int, float, list, dict, tuple, set)
  - Control flow (if, for, while, break/continue)
  - Functions (default args, keyword args, return values)
  - Modules & imports (math, os, sys)
- File handling:
  - Reading/writing text, CSV, and Excel files
  - Using pathlib for file paths
- Clinical trial data overview:
  - Common dataset types (demographics, AE, visits, labs)
  - Understanding CDISC SDTM dataset naming conventions
- Using Pandas for basic operations:
  - Loading datasets
  - Viewing structure & summary statistics
  - Selecting & filtering rows
  - Sorting data

Lab: Load **Subject Demographics** file, clean missing Age/Sex/Country, and produce patient count by country & gender.

Day 2 – Advanced Python Concepts for Automation

Topics:

- Advanced functions:
  - Lambda functions, map, filter, reduce
  - List/dict comprehensions
- Object-Oriented Programming:
  - Creating classes & methods
  - Inheritance & method overriding
  - Practical OOP for automation tools
- Creating reusable modules:
  - Writing helper functions for repeated validations
  - Structuring project folders
- Error handling & logging:
  - try/except/finally
  - Logging data validation results for audit trails
- Using Python virtual environments for project isolation

Lab: Create a **Clinical Data Validator** class that reads lab results, flags abnormal values per reference range, and logs errors.

Day 3 – Data Processing & Specification Validation

Topics:

- Advanced Pandas:
  - GroupBy and multiple aggregations
  - Pivot tables for trial summaries
  - Merging datasets (inner, left, right joins)
- Data validation techniques:
  - Detecting missing patient visits
  - Verifying variable names match CRF specifications
  - Data type enforcement
- Comparing datasets:
  - Between CRF specs & actual data
  - Highlighting discrepancies
- Creating discrepancy reports for stakeholders
- Using openpyxl to style Excel output

Lab: Compare a **Vital Signs specification** to collected data, identify missing variables & mismatches, and create a compliance summary.

Day 4 – Automation & Reporting

Topics:

- Automating multi-file processing:
  - Iterating over site-submitted datasets
  - Consolidating into a master file
- Automating periodic tasks:
  - Scheduling scripts (Windows Task Scheduler / cron jobs)
- Creating summary reports:
  - Adding charts with xlsxwriter
  - Exporting to Excel and PDF
- Data visualization for clinical metrics:
  - Patient enrollment trends
  - AE severity breakdowns
  - Visit compliance heatmaps
- Introduction to **Streamlit** dashboards:
  - Adding interactivity
  - Filtering and searching patient records

Lab: Automate reading monthly site submissions, validate completeness, merge into master dataset, and export multi-tab Excel + PDF reports.

Day 5 – Hands-On Workshop: Build a Clinical Data Automation Tool

Topics:

- Identifying automation opportunities in clinical workflows
- Defining problem statements
- Breaking down requirements into Python tasks
- Integrating:
    - Data loading
    - Cleaning & validation
    - Reporting & visualization
- Using APIs for clinical reference data:
    - Fetching MedDRA codes
    - Drug info lookups
- Data anonymization techniques for PHI protection
- Final best practices:
    - Code documentation
    - Reusability & maintainability