

Document Generated: 06/30/2024

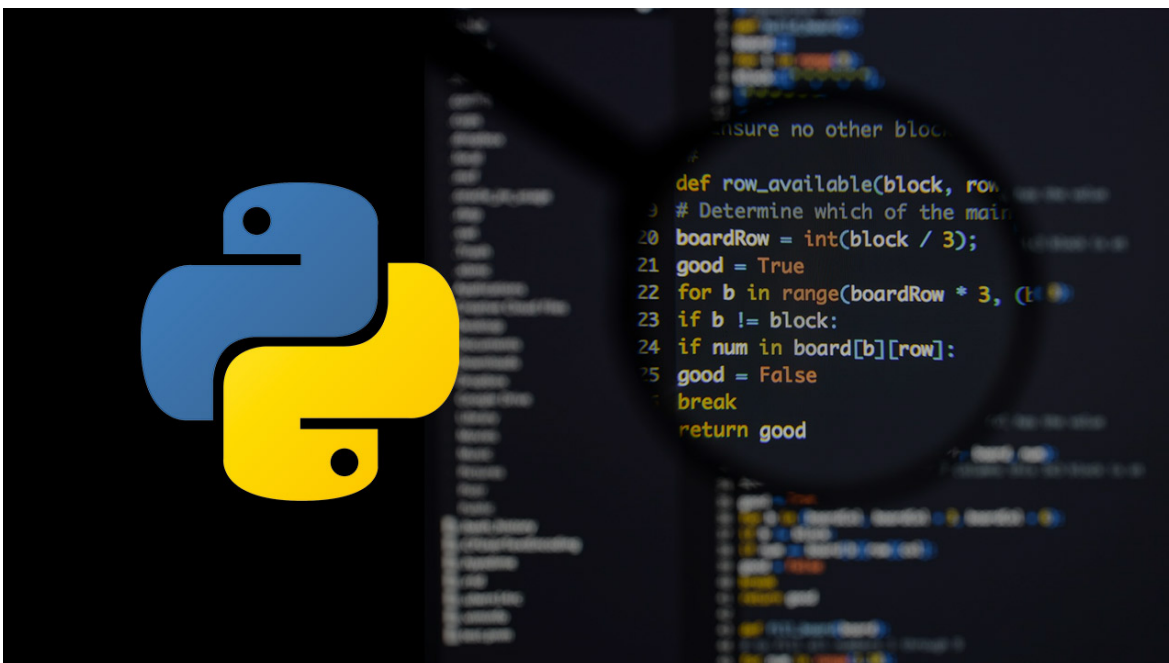
Learning Style: Virtual Classroom

Provider:

Difficulty: Advanced

Course Duration: 4 Days

## Advanced Python Programming (TTPS4850)



### About this Course:

This is an Advanced-Level Training Program specifically designed for Experienced Python Programmers striving to learn the advanced Python Programming concepts. This practical training program covers both the intermediate and advanced topics and helps professionals develop a conceptual understanding of working with key Python features and functionalities. On average, a professional Python Programmer earns \$111,557 annually.

This course is designed to help professionals leverage the OS tools and services to augment the performance of Python-based Programs and Applications. Professionals get to learn about Code Graphical Interfaces, Query Databases, XML

Data Processing, Static & Classic Data Methods, Module Creation & Execution, File System Essentials, and Python Program Structure. This course is not simply an academic overview of Python grammar and Syntax and provides a comprehensive practical experience of working with Python.

## **Course Objectives:**

The core objective of this course is to help professionals gain a better understanding and sound knowledge of the following key concepts:

- Python Data Types, Mapping, Sequences, and Program Structure
- File System Essentials and External Commands Application
- Operating System Services Application and Benefits
- Pythonic Programming Overview and Lambda Functions and Idioms
- Working with Code Graphical Interfaces
- Module Creation and Evaluation with Unit Tests
- Module Execution, Code Initialization, and Name & Package Resolution
- Network Services Interaction and Classes Essentials
- Static & Classic Data Methods and Classes Inheritance
- Getting to know Query Databases and XML Data Processing
- Metaprogramming, Monkey Patching, and Module Inspection
- Program Analysis and Benchmarking & Profiling
- Formulating Installers and Distribution Concepts
- PyQT4 GUI Programming and Python Extension

## **Audience:**

- Experienced Python Programmers
- IT Professionals & Experts
- Professional Web Developers

## **Prerequisites:**

Professionals planning to enroll in the Advanced Python Programming (TTPS4850) Course must comply with the following prerequisites:

- Introduction to Python Programming (TTEY101) Certificate or Equivalent Knowledge
- Essential Python Programming (TTPS4810) Certificate or Equivalent Knowledge

## **Course Outline:**

### **Module 1: Python refresher**

Data types  
Sequences  
Mapping types  
Program structure

- Files and console I/O
- Conditionals
- Loops
- Built-ins

## **Module 2: OS services**

- The OS module
- Environment variables
- Launching external commands
- Walking directory trees
- Paths, directories, and filenames
- Working with file systems
- Dates and times

## **Module 3: Pythonic programming**

- The Zen of Python
- Common idioms
- Lambda functions
- List comprehensions
- Generator expressions
- String formatting

## **Module 4: Modules and packages**

- Initialization code
- Namespaces
- Executing modules as scripts
- Documentation
- Packages and name resolution
- Naming conventions
- Using imports

## **Module 5: Classes**

- Defining classes
- Instance methods and data
- Properties
- Initializers
- Class and static methods/data
- Inheritance

## **Module 6: Metaprogramming**

- Implicit properties
- globals() and locals()
- Working with attributes
- The inspect module
- Decorators

Monkey patching

## **Module 7: Programmer tools**

Analyzing programs  
Using pylint  
Testing code  
Using unittest  
Debugging  
Profiling and benchmarking

## **Module 8: Distributing modules**

Distribution concepts  
setuptools  
Creating setup.py  
Building installers  
Running installers

## **Module 9: Database access**

The DB API  
Available Interfaces  
Connecting to a server  
Creating and executing a cursor  
Fetching data  
Parameterized statements  
Metadata  
Transaction control  
Other DBMS modules

## **Module 10: GUI programming with PyQt4**

About QT4  
Getting started with the designer  
Widget properties  
Predefined dialogs  
Generating the UI  
Wiring up events  
Advanced Topics

## **Module 11: Network programming**

Sockets  
Clients  
Servers  
Application protocols  
Forking servers  
Binary data  
The struct module

## **Module 12: Threads**

- When to use threads?
- The Global Interpreter Lock
- The threading module
- Simple threading
- Sharing variables
- Threaded servers
- The queue module
- Debugging threaded programs
- Alternatives to threading

## **Module 13: XML and JSON**

- Working with XML
- DOM and Sax
- Introducing ElementTree and lxml
- Parsing XML
- Navigating the document
- Creating a new XML document
- JSON
- Parsing JSON into Python
- Converting Python into JSON

## **Module 14: Extending Python**

- About non-Python modules
- Overview of a C extension
- Writing C by hand
- Loading modules with ctypes

## **Module 15: Subprocesses**

- Running external commands with subprocess
- Getting command status
- Managing STDOUT, STDERR, and STDIN
- The sh module (non-Windows systems only)
- Creating a simple command
- Keyword arguments
- Running commands in the background
- Piping and redirection
- Working with STDIO
- Exit codes
- Advanced features

## **Credly Badge:**

**Display your Completion Badge And Get The**



## Recognition You Deserve.

Add a completion and readiness badge to your LinkedIn profile, Facebook page, or Twitter account to validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

[Find Out More](#) or [See List Of Badges](#)