

Course Outline

Advanced Python Programming / Next-Level Python Course TTPS4850: 4 days Instructor Led

About this course

Advanced Python Programming is a practical, hands-on Python training course that thoroughly explores intermediate to advanced level topics and skills, with a focus on enterprise development. Throughout the course, students will learn how to Leverage OS services, Code graphical interfaces for applications, create modules and run unit tests, define classes, interact with network services, query databases, process XML data, and much more. This comprehensive, practical course provides an in-depth exploration of working with the programming language, not an academic overview of syntax and grammar.

Audience profile

This is an **intermediate and beyond level** Python course geared for students experienced with Python who want to use Python in web development projects or automate or simplify common tasks with the use of Python scripts. Basic incoming practical experience working with Python is required, along with a working, user-level knowledge of Unix/Linux, Mac, or Windows. This course does not cover Python fundamentals.

At course completion

After completing this course, students will be able to:

- Leverage OS services
- Add enhancements to classes
- Code graphical interfaces for applications
- Understand advanced Python metaprogramming concepts
- Create easy-to-use and easy-to-maintain modules and packages
- Implement and run unit tests
- Create multithreaded and multi-process applications
- Interact with network services
- Design professional scripts
- Query databases
- Process XML, CSV, and JSON data
- Working with more data types if time permits
- Using type hints if time permits

Course Outline

1. Python Quick Refresher

- Builtin data types
- Lists and tuples
- Dictionaries and sets
- Program structure
- Files and console I/O
- If statement
- *for* and *while* loops

2. OS Services

Course Outline

- The os and os.path modules
 - Environment variables
 - Launching external commands with subprocess
 - Walking directory trees
 - Paths, directories, and filenames
 - Working with file systems
3. **Dates and Times**
- Basic date and time classes
 - Different time formats
 - Converting between formats
 - Formatting dates and times
 - Parsing date/time information
4. **Binary Data**
- What is Binary Data?
 - Binary vs text
 - Using the Struct module
5. **Pythonic Programming**
- The Zen of Python
 - Tuples
 - Advanced unpacking
 - Sorting
 - Lambda functions
 - List comprehensions
 - Generator expressions
 - String formatting
6. **Functions, modules, and packages**
- Four types of function parameters
 - Four levels of name scoping
 - Single/multi dispatch
 - Relative imports
 - Using `__init__` effectively
 - Documentation best practices
7. **Intermediate classes**
- Class/static data and methods
 - Inheritance (or composition)
 - Abstract base classes
 - Implementing protocols (context, iterator, etc.) with special methods

Course Outline

8. Metaprogramming

- Implicit properties
- globals() and locals()
- Working with object attributes
- The inspect module
- Callable classes
- Decorators
- Monkey patching

9. Developer Tools

- Analyzing programs with pylint
- Using the debugger
- Profiling code
- Testing speed with benchmarking

10. Unit testing with PyTest

- What is a unit test?
- Writing tests
- Working with fixtures
- Test runners
- Mocking resources

11. Database access

- The DB API
- Available Interfaces
- Connecting to a server
- Creating and executing a cursor
- Fetching data
- Parameterized statements
- Using Metadata
- Transaction control
- ORMs and NoSQL overview

12. PyQt

- Overview
- Qt Architecture
- Using designer
- Standard widgets
- Event handling
- Extras

13. Network Programming

Course Outline

- Builtin classes
- Using requests
- Grabbing web pages
- Sending email
- Working with binary data
- Consuming RESTful services
- Remote access (SSH)

14. Multiprogramming

- The threading module
- Sharing variables
- The queue module
- The multiprocessing module
- Creating pools
- About async programming

15. Scripting for System Administration

- Running external programs
- Parsing arguments
- Creating filters to read text files
- Logging

16. Serializing data – XML and JSON

- Working with XML
- XML modules in Python
- Getting started with ElementTree
- Parsing XML
- Updating an XML tree
- Creating a new document
- About JSON
- Reading JSON
- Writing JSON
- Reading/writing CSV files
- YAML, other formats as time permits

Time Permitting Sessions

17. Advanced data handling

- Discover the collections module
- Use defaultdict, Counter, and namedtuple
- Create dataclasses
- Store data offline with pickle

18. Type hinting

Course Outline

- Annotate variables
- Learn what type hinting does NOT do
- Use the typing module for detailed type hints
- Understand *union* and *optional* types
- Write stub interfaces