

## PY-INTP

## PYTHON ASSOCIATE - PROGRAMMER I (v3)



DURATION	LEVEL	TECHNOLOGY	DELIVERY METHOD	TRAINING CREDITS
5 Days	Intermediate	Python	Instructor Led	NA

### INTRODUCTION

This 5-day course covers some Python introduction topics in more detail, and adds many new ones, with a focus on enterprise development. This is a hands-on programming class. All concepts are reinforced by informal practice during the lecture followed by lab exercises. Many labs build on earlier labs, which helps students retain the earlier material.

### AUDIENCE PROFILE

This course is intended for advanced users, system administrators and website administrators who want to use Python to support their server installations, as well as anyone else who wants to automate or simplify common tasks with the use of Python scripts.

### PREREQUISITES

Before attending this course, students must have:

- Be able to write simple Python scripts, using basic data types, program structures, and the standard Python library.

### COURSE OBJECTIVES

After completing this course, students will be able to:

- Utilize variables, data types, operators, and control flow structures effectively
- Use the various pythonic programming principles
- Understand the use of various modules and packages
- Develop well-structured functions and modules
- Implement Metaprogramming
- Use Python developer tools
- Access databases using Python programming
- Load, clean, and manipulate data using Pandas
- Understand and use network programming
- Use Python programming for System Administration and Scripting
- Understand and use XML and JSON

### COURSE CONTENT

#### Module 1: Python Refresher

- Variables
- Basic Python Data Types
- Sequence Types
- Mapping Types
- Program Structure
- Files and Console I/O
- Conditionals
- Loops
- Built-ins
- Functions
- Modules
- Packages

#### Module 2: OS Services

- The OS Module
- Paths, Directories and Filenames
- Environment Variables
- Launching external Programs
- Walking Directory Trees

#### Module 3: Dates and Times

- Programming
- Python modules for dates and times
- Ways to store dates and times
- Formatting dates and times
- Parsing date/time strings
- Parsing dates the easier way
- Converting dates and times
- Time zones
- Generating calendars

#### Module 4: Binary Data

- “Binary” (raw, or non-delimited) data
- Binary vs Text data
- Using Struct
- Bitwise operations

#### Module 5: Python Programming

- The Zen of Python
- Tuples
- Iterable unpacking
- Unpacking function arguments
- The sorted() function
- Custom sort keys
- Lambda functions
- List comprehensions
- Dictionary comprehensions
- Set comprehensions
- Iterables
- Generator Expressions
- Generator functions
- String formatting
- f-strings

#### Module 6: Functions, Modules and Packages

- Functions

- Function parameters
- Default parameters
- Python Function parameter behavior (from PEP3102)
- Name resolution (AKA Scope)
- The global statement
- Modules
- Using import
- How import \*can be dangerous
- Module search path
- Executing modules as scripts
- Packages
- Configuring import with `__init__.py`
- Documenting modules and packages
- Python style

## Module 7: Intermediate Classes

- What is a class?
- Defining Classes
- Object Instances
- Instance attributes
- Instance Methods
- Constructors
- Getters and setters
- Properties
- Class Data
- Class Methods
- Inheritance
- Using `super()`
- Multiple Inheritance
- Abstract base classes
- Special Methods
- Static Methods

## Module 8: Metaprogramming

- Metaprogramming
- `Globals()` and `locals()`
- The `inspect` module
- Working with attributes
- Adding instance methods
- Decorators
- Applying decorators
- Trivial Decorator
- Decorator functions
- Decorator Classes
- Decorator parameters
- Creating classes at runtime
- Monkey Patching
- Callable classes
- Do you need a Metaclass?
- About metaclass
- Mechanics of a metaclass
- Singleton with a metaclass

## Module 9: Developer Tools

- Program development
- Comments
- Pylint
- Customizing pylint
- Using `pyreverse`
- The Python debugger
- Starting debug mode
- Stepping through a program
- Setting breakpoints
- Profiling
- Benchmarking

## Module 10: Unit Tests with pytest

- What is a unit test?
- The `pytest` module
- Creating tests
- Running tests (basics)
- Special assertions
- Fixtures
- User-defined fixtures
- Builtin fixtures
- Configuring fixtures
- Parametrizing tests
- Marking tests
- Running tests (advanced)
- Skipping and failing
- Mocking data
- Pymock objects
- Pytest and Unittest

## Module 11: Database Access

- The DB API
- Connecting to a Server
- Creating a Cursor
- Executing a Statement
- Fetching Data
- SQL Injection
- Parameterized Statements
- Dictionary Cursors
- Metadata
- Transactions
- Object-relational Mappers
- NoSQL

## Module 12: PyQt

- What is PyQt?
- Event Driven Applications
- External Anatomy of a PyQt Application
- Internal Anatomy of a PyQt Application
- Using designer
- Designer-based application workflow
- Naming conventions
- Common Widgets
- Layouts
- Selectable Buttons
- Actions and Events
- Signal/Slot Editor
- Editing modules
- Menu Bar
- Status Bar
- Forms and validation
- Using Predefined Dialogs
- Tabs
- Niceties
- Working with Images
- Complete Example

## Module 13: Network Programming

- Grabbing a web page
- Consuming Web services
- HTTP the easy way
- Sending e-mail
- Email attachments
- Remote Access
- Copying files with Paramiko

## Module 14: Multiprogramming

- Multiprogramming
- What are Threads?

- The Python Thread Manager
- The `threading` Module
- Threads for the impatient
- Creating a thread class
- Variable sharing
- Using queues
- Debugging threaded Programs
- The `multiprocessing` module
- Using pools
- Alternatives to multiprogramming

## Module 15: Effective Scripts

- Using `glob`
- Using `shlex.split()`
- The `subprocess` module
- Subprocess convenience function
- Capturing a `stdout` and `stderr`
- Permissions
- Using `Shutil`
- Creating a useful command line script
- Creating filters
- Parsing the command line
- Simple Logging
- Formatting log entries
- Logging exception information
- Logging to other destinations

## Module 16: Serializing Data

- About XML
- Normal Approaches to XML
- Which module to use?
- Getting started With `ElementTree`
- How `ElementTree` Works
- `Element`
- Creating a New XML Document
- Parsing An XML Document
- Navigating the XML Document
- Using `Xpath`
- About JSON
- Reading JSON
- Writing JSON
- Customizing JSON
- Reading and Writing YAML
- Reading CSV data
- Nonstandard CSV
- Using `csv.DictReader`
- Writing CSV Data
- Pickle

## Module 17: Advanced Data Handling

- Deep vs Shallow copying
- Default dictionary values
- Counting with `Counter`
- Named Tuples
- Printing data structures
- Zipped archives
- Serializing Data

## Module 18: Type Hinting

- Type Hinting
- Static Analysis Tools
- Typing Module
- Input Types
- Variance
- Union and Optional
- Stub Type Hinting

### ASSOCIATED CERTIFICATIONS & EXAM

There is no associated exam for this course.