

## PY-INTP



## PYTHON ASSOCIATE - PROGRAMMER I

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 web site 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:

- Understand the OS Services
- Use the various pythonic programming principles
- Understand the use of various modules and packages
- Define and use Classes
- Implement Metaprogramming
- Use Python developer tools
- Access databases using Python programming
- Use PyQt4 framework
- 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
- Sequences
- Mapping Types
- Program Structure
- Files and Console I/O
- Conditionals
- Loops
- Built-ins

#### Module 2: OS Services

- The OS Module
- Paths, Directories and Filenames

- Environment Variables
- Launching external Programs
- Walking Directory Trees
- The Datetime Module
- The Calendar Module

#### Module 3: Pythonic

- Programming
- The Zen of Python
- Common Python Idioms
- Unpacking Function Arguments
- Lambda Functions - List Comprehensions

- Iterables
- Writing Generator
- String Tricks
- String Formatting

#### Module 4: Modules and Packages

- Using import
- Module Search Path
- Namespaces
- Executing Modules as Scripts
- Packages
- Configuring Import with `_init__.py`

- Name Resolution (AKA scope)
- Python Style

## Module 5: Classes

### Defining Classes

- Instance Objects
- Instance Attributes
- Instance Methods
- `__init__`
- Properties
- Class Data
- Class Methods
- Inheritance
- Multiple Inheritance
- Using `Super()`
- Special Methods
- Class-Private -Variables
- Static Methods

## Module 6: Metaprogramming

- `Globals()` and `Locals()`
- Working with Attributes
- The `Inspect` Module
- Decorator Functions
- Decorator Classes
- Decorator Parameters
- Creating Classes at Runtime
- Monkey Patching

## Module 7: Developer Tools

- Program Development
- Comments
- Pylint
- Customizing pylint
- Using `pyreverse`
- The `unittest` module
- Fixtures
- Skipping Tests
- Making a Suite of Tests
- Automated Test Discovery
- Using `Nose`
- The Python Debugger
- Starting Debug Mode
- Stepping Through a Program
- Setting Breakpoints
- Profiling
- Benchmarking

## Module 8: Database Access The DB API

- Available Interfaces
- Connecting to a Server
- `connect()` examples

- Creating a cursor
- Executing a statement
- Parameterized statements
- Dictionary cursors
- Metadata Transactions
- Object-relational mappers

## Module 9: PyQt4

- Event Driven Applications
- Gui Application Flow Chart
- External Anatomy of a qt4 Application
- Internal Anatomy of a qt4 Application
- Using Designer
- Anatomy of a Designer Based Application
- Naming Conventions
- Common Widgets
- Layouts
- Selectable Buttons
- Actions and Events
- Menu Bar
- Status Bar
- Using Predefined - Dialogs
- Creating Custom - Dialogs
- Tabs
- Niceties
- Working with Images
- Complete Example

## Module 10: Network Programming

- Sockets
- Socket options
- Client Concepts
- Server Concepts
- Application Protocols
- Forking Servers
- Grabbing HTML from the Web
- Consuming Web Services
- Web Data the Easier Way
- Sending email
- Binary Data
- The `struct` module

## Module 11: 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
- Alternatives to multiprocessing

## Module 12: System Administration and Scripting

- The subprocess module
- Subprocess Convenience
- Functions Using the module
- Permissions
- Saving Information
- Creating a useful Command Line Script
- Creating Filters
- Parsing the Command Line
- Simple logging
- Logging Levels
- Formatting Log Entries
- Logging to other Destinations

## Module 13: XML and JSON About XML

- Normal approaches to XML
- Which Module to use?
- Getting Started with `ElementTree`
- How `ElementTree` works
- Creating a new XML Document
- Parsing an XML Document
- Navigating the XML Document
- Using XPath
- About JSON
- Reading JSON
- Writing JSON

## Module 14: Extending Python

- Why extend Python?
- Ways to extend Python with C
- Hand-Coded C
- Overview
- The C Program
- Methods
- The Method Table
- The `Init` function
- Handling errors
- Custom Exception -Objects
- Putting it all together
- Using SWIG
- The interface file
- Generating the Wrappers
- Building and installing the ex

## ASSOCIATED CERTIFICATIONS & EXAM

None