Python Essentials

Course Duration: 3-day; Instructor-led
Time Schedule: 9am-5pm
Lunch: 1:00pm- 2:00pm
Morning Tea-break: 10:30am-10:45am
Afternoon Tea-break: 3:30pm -3:45pm

WHAT YOU WILL LEARN
This course introduces the student to the Python language. On completion of this class the student should feel comfortable with writing Python programs. The course provides insight to the features of Python that make it an excellent choice for projects of virtually any size

AUDIENCE
This course is designed for those already using Korn shell scripting, Perl, or C programming languages to manipulate files or control processes

PREREQUISITES
This course requires participants to meet the following prerequisites:

- Introduction to Unix course or equivalent experience
- Recent programming experience

METHODOLOGY
This program will be conducted through Instructor-led (classroom)

COURSE OBJECTIVES
After completing this course, students will be able to:
Write Python programs dealing with sequences and mappings, program flow control, system calls and file manipulation

COURSE OUTLINE

Module 1

Why Python?

- Why do people use Python?
- Is Python a scripting language?
- What can I do with Python?
- What are Python's technical strengths?

How's Python Runs Programs?

- Introducing the Python interpreter
- Program execution
- Execution model variations

Numbers

- Python program structure
- Why use built-in types?
- Numbers
- Python expression Operators
- Numbers in action
- The dynamic typing interlude

Strings

- String literals
- Strings in action
- String formatting
- String methods
- General type categories

List & Dictionaries

- Lists
- Lists in action
- Dictionaries
- Dictionaries in action

Tuples, Files and Everything Else

- Tuples
- Files
- Type categories revisited
- Object generality
- References versus copies
- Comparisons, equality, and truth
- Python's type hierarchies
- Other types in Python
- Built-in type gotchas

Modules 2

Assignment, Expressions & Print

- Assignment statements
- Expression statements
- Print statements

If Test

- If statements
- Python syntax rules
- Truth tests

While and For Loops

- While loops
- Break, continue, pass, and the loop else
- For loops
- Loop variations

Function Basics

- Why use functions?
- Coding functions
- Definition & calls

- Intersecting sequences

Scope & Arguments

- Scope rules
- The global statement
- Scopes and nested functions
- Passing arguments
- Special argument matching modes

Module 3

The Big Pictures

- Why use modules?
- Python program architecture
- How imports work?

Coding Basics

- Module creation
- Module usage
- Module namespaces
- Reloading modules

Packages

- Package import basics
- Package import example
- Why use package imports?

OOP: The Big Picture

- Why use classes?
- OOP from 30,000 feet

Class Coding Basics

- Classes generate multiple instance objects
- Classes are customized by inheritance
- Classes can intercept python operators

Module 4

Class Coding Details

- The class statement
- Methods
- Inheritance
- Operator overloading
- Namespaces

Designing With Classes

- Python and OOP
- Classes as records
- OOP and inheritance: "is-a" relationships
- OOP and composition: "has-a" relationships
- OOP and delegation
- Multiple inheritance
- Classes are objects: Generic object factories
- Methods are objects: Bound or unbound

- Classes versus modules

Exception Basic

- Why use exceptions?
- Exception handling: The short story
- The try/except/else statement
- The try/finally statement
- The raise statement
- The assert statement

Exception Objects

- String-based exceptions
- Class-based exceptions
- General raise statement forms

Designing with Exceptions

- Nesting exception handlers
- Exception idioms
- Exception design tips
- Exception gotchas
- Core language summary

Module 5

Common Tasks in Python

- Exploring on your own
- Conversions, numbers and comparisons
- Manipulating strings
- Data structure manipulations
- Manipulating files and directories
- Internet-related modules
- Executing programs
- Debugging, testing, timing, profiling

Advance Topics

- Web frameworks
- GUI frameworks
- Content management frameworks