Course Name : **PYTHON INTERMEDIATE TO ADVANCED**
Duration : **3 Days (Physical Classroom / Virtual Live Instructor)**
Skill Level : **Intermediate**

COURSE DESCRIPTION:
The "Python Intermediate to Advanced" course is designed to elevate programmers from proficiency to expertise by delving into Python's capabilities comprehensively. Beginning with foundational data structures like lists and tuples, the curriculum emphasizes creation, manipulation, and advanced iteration techniques. participants progress to more intricate concepts such as dictionaries and sets, where they learn about key-value management, set operations, and the utility of frozensets for immutability. The course also covers string manipulation, including formatting and understanding immutability principles, before moving on to specialized topics like itertools and lambda functions for efficient and concise programming solutions.

Building on these fundamentals, the course addresses critical aspects of robust software development such as error handling and logging. It includes mastering exception handling, creating custom exceptions, and implementing comprehensive logging practices. Students also gain proficiency in JSON handling for data interchange and using decorators to dynamically modify functions and methods. Advanced topics include memory-efficient programming with generators, optimizing performance through multithreading and multiprocessing, and effective management of function arguments and context managers. By exploring Python's memory management and interpreter workings, the course ensures learners are well-prepared to tackle complex programming challenges with confidence and skill.

WHAT WILL YOU LEARN?
In the "Python Intermediate to Advanced" course, you'll master advanced operations with lists, tuples, dictionaries, and sets, and gain proficiency in string formatting and immutability. You'll work with specialized collections, itertools, and lambda functions, enhance your error handling and logging skills, and manage JSON data interchange. You'll also gain insights into Python's memory management and interpreter workings, preparing you for complex programming challenges.

PREREQUISITE:
Basics Python/Programming experience is required.

METHODOLOGY:
This program will be conducted with interactive lectures, PowerPoint presentations, discussions, and practical exercises. This course can be conducted as instructor-led (ILT) or virtual instructor-led training (VILT).

JOB SCOPE:
Upon completion of this course, candidates may pursue the following career paths:

- Software Engineer/Developer
- Data Scientist
- AI/Machine Learning Engineer
- Data Engineer
- DevOps Engineer

Rynet Malaysia Sdn. Bhd. (899837-K)
95-2, Block H, The Suites, Jaya One, 72A, Jalan Universiti, 46200 Petaling Jaya,Selangor, Malaysia.
Phone: +603-7623-1088 | Fax: +603-7660-9393

MODULE 1: PYTHON CRASH COURSE

- Welcome
- Introduction to Python
- Setting Up the Environment
- Basic Syntax and Data Types
- Control Structures (If-Else, Loops)
- Functions
- Input and Output
- Basic File Operations

MODULE 2: LISTS

- Creating a list
- Access elements
- Change items
- Useful methods
- Copy a list
- Iterating
- Check if an item exists
- Slicing
- List comprehension
- Nested lists

MODULE 3: TUPLES

- Reasons to use a tuple over a list
- Create a tuple
- Access elements
- Add or change items
- Delete a tuple
- Iterating
- Check if an item exists
- Usefule methods
- Slicing
- Unpack tuple
- Nested tuples
- Compare tuple and list

MODULE 4: DICTIONARIES

- Create a dictionary
- Access items
- Add and change items
- Delete items
- Check for keys
- Looping through dictionary
- Copy a dictionary

Rynet Malaysia Sdn. Bhd. (899837-K)
95-2, Block H, The Suites, Jaya One, 72A, Jalan Universiti, 46200 Petaling Jaya,Selangor, Malaysia.
Phone: +603-7623-1088 | Fax: +603-7660-9393

MODULE 5: SETS

- Create a set
- Add elements
- Remove elements
- Check if element is in Set
- Iterating
- Union and Intersection
- Difference of sets
- Updating sets
- Copying
- Subset, Superset, and Disjoint
- Frozenset

MODULE 6: STRINGS

- Creation
- Access characters and substrings
- Concatenate two or more strings
- Iterating
- Check if a character or substring exists
- Useful methods
- Format
- f-Strings
- More on immutability and concatenation

MODULE 7: COLLECTION

- Counter
- namedtuple
- OrderedDict
- defaultdict
- deque

MODULE 8: ITERTOOLS

- product()
- permutations()
- combinations() and combinations_with_replacement()
- accumulate()
- groupby()

MODULE 9: LAMBDA FUNCTIONS

- Usage example: Lamdba inside another function
- Custom sorting using a lambda function as key parameter
- Use lambda for map function
- Use lambda for filter function
- Reduce

Rynet Malaysia Sdn. Bhd. (899837-K)
95-2, Block H, The Suites, Jaya One, 72A, Jalan Universiti, 46200 Petaling Jaya,Selangor, Malaysia.
Phone: +603-7623-1088 | Fax: +603-7660-9393

MODULE 10: EXCEPTIONS AND ERROR HANDLING

- Syntax Errors
- Exceptions
- Raising an Exception
- Handling Exceptions
- else clause
- finally clause
- Common built-in Exceptions
- Define your own Exceptions

MODULE 11: LOGGING

- Log Level
- Configuration
- Logging in modules and logger hierarchy
- Propagation
- LogHandlers
- Example of a filter
- Other configuration methods
- Capture Stack traces
- Rotating FileHandler
- TimedRotatingFileHandler
- Logging in JSON Format

MODULE 12: JSON

- JSON format
- From Python to JSON
- FROM JSON to Python
- Working with Custom Objects
- Template encode and decode functions

MODULE 13: DECORATORS

- Function decorators
- The decorator syntax
- What about function arguments
- Return values
- What about the function identity?
- The final template for own decorators
- Decorator function arguments
- Nested Decorators
- Class decorators
- Some typical use cases

Rynet Malaysia Sdn. Bhd. (899837-K)
95-2, Block H, The Suites, Jaya One, 72A, Jalan Universiti, 46200 Petaling Jaya,Selangor, Malaysia.
Phone: +603-7623-1088 | Fax: +603-7660-9393

MODULE 14: GENERATORS

- Execution of a generator function
- Big advantage: Generators save memory!
- Another example: Fibonacci numbers
- Generator expressions
- Concept behind a generator

MODULE 15: MULTITHREADING

- Create and run threads
- Share data between threads
- How to use Locks
- Race condition
- Avoid race conditions with Locks
- Use the lock as a context manager
- Using a queue in multithreading

MODULE 16: MULTIPROCESSING

- Create and run processes
- Share data between processes
- How to use Locks
- Race condition
- Avoid race conditions with Locks
- Use the lock as a context manager
- Using Queues in Python
- Using a queue in multiprocessing

MODULE 17: FUNCTION ARGUMENTS

- Arguments and parameters
- Positional and keyword arguments
- Default arguments
- Variable-length arguments (*args and **kwargs)
- Forced keyword arguments
- Unpacking into agruments
- Local vs global variables
- Parameter passing

MODULE 18: THE ASTERISK (*) OPERATOR

- Multiplication and power operations
- Creation of list, tuple, or string with repeated elements
- *args , **kwargs , and keyword-only arguments
- Unpacking for function arguments
- Unpacking containers
- Merge iterables into a list / Merge dictionaries

Rynet Malaysia Sdn. Bhd. (899837-K)
95-2, Block H, The Suites, Jaya One, 72A, Jalan Universiti, 46200 Petaling Jaya,Selangor, Malaysia.
Phone: +603-7623-1088 | Fax: +603-7660-9393

MODULE 19: SHALLOW vs DEEP COPYING

- Shallow vs Deep Copying
- Assignment operation
- Shallow copy
- Deep copies
- Custom objects

MODULE 20: CONTEXT MANAGERS

- Examples of context managers
- Implementing a context manager as a class
- Handling exceptions
- Implementing a context manager as a generator

MODULE 21: PYTHON ADVANCED

- Higher Order Functions
- Lambda
- Sorting with Key
- Map, Filter, Reduce
- List Comprehension
- Dictionary Comprehension
- Zip Function
- MultiThreading
- How Python Interpreter Works?
- Memory Management

CONCLUSION

- QA
- Useful References and Books
- Feedback

Rynet Malaysia Sdn. Bhd. (899837-K)
95-2, Block H, The Suites, Jaya One, 72A, Jalan Universiti, 46200 Petaling Jaya,Selangor, Malaysia.
Phone: +603-7623-1088 | Fax: +603-7660-9393