PYTHON PROGRAMMING (5 DAYS – 4 HRS PER DAY)

By Dr. Vishwanath Rao

Day 1

Introduction to Languages
* What is Language?
* Types of languages
* Introduction to Translators
  * Compiler
  * Interpreter
* What is Scripting Language?
* Types of Script
* Programming Languages v/s Scripting Languages
* Difference between Scripting and Programming languages
* What is programming paradigm?
* Procedural programming paradigm
* Object Oriented Programming paradigm

Introduction to Python
* What is Python?
* WHY PYTHON?
* History
* Features – Dynamic, Interpreted, Object oriented, Embeddable, Extensible, Large standard libraries, Free and Open source
* Why Python is General Language?
* Limitations of Python
* What is PSF?
* Python implementations
* Python applications
* Python versions
* PYTHON IN REALTIME INDUSTRY
* Difference between Python 2.x and 3.x
* Difference between Python 3.7 and 3.8
* Software Development Architectures

Python Software's
* Python Distributions
* Download &Python Installation Process in Windows, Unix, Linux and Mac
* Online Python IDLE
* Python Real-time IDEs like Spyder, Jupyter Note Book, PyCharm, Rodeo, Visual Studio Code, ATOM, PyDevetc

Python Language Fundamentals
* Python Implementation Alternatives/Flavors
* Keywords
* Identifiers
* Constants / Literals
* Data types
* Python VS JAVA

* Python Syntax
Different Modes of Python
* Interactive Mode
* Scripting Mode
* Programming Elements

Day 2

- Structure of Python program
  * First Python Application
  * Comments in Python
  * Python file extensions
  * Setting Path in Windows
  * Edit and Run python program without IDE
  * Edit and Run python program using IDEs
  * INSIDE PYTHON
  * Programmers View of Interpreter
  * Inside INTERPRETER
  * What is Byte Code in PYTHON?
  * Python Debugger
  Python Variables
  * bytes Data Type
  * byte array
  * String Formatting in Python
  * Math, Random, Secrets Modules
  * Introduction
  * Initialization of variables
  * Local variables
  * Global variables
  * ‘global’ keyword
  * Input and Output operations
  * Data conversion functions – int(), float(), complex(), str(), chr(), ord()
-
- Operators
  * Arithmetic Operators
  * Comparison Operators
  * Python Assignment Operators
  * Logical Operators
  * Bitwise Operators
  * Shift operators
  * Membership Operators
  * Identity Operators
  * Ternary Operator
  * Operator precedence
  * Difference between “is” vs “==”
  Input & Output Operators
  * Print
  * Input

* Command-line arguments

Control Statements

* Conditional control statements
* If
* If-else
* If-elif-else
* Nested-if
* Loop control statements
* for
* while
* Nested loops
* Branching statements
* Break
* Continue
* Pass
* Return
* Case studies

Day 3

- Data Structures or Collections
* Introduction
* Importance of Data structures
* Applications of Data structures
* Types of Collections
* Sequence
* Strings, List, Tuple, range
* Non sequence
* Set, Frozen set, Dictionary
* Strings
* What is string
* Representation of Strings
* Processing elements using indexing
* Processing elements using Iterators
* Manipulation of String using Indexing and Slicing
* String operators
* Methods of String object
* String Formatting
* String functions
* String Immutability
* Case studies

List Collection

* What is List
* Need of List collection
* Different ways of creating List
* List comprehension
* List indices
* Processing elements of List through Indexing and Slicing
* List object methods

* List is Mutable
* Mutable and Immutable elements of List
* Nested Lists
* List_of_lists
* Hardcopy, shallowCopy and DeepCopy
* zip() in Python
* How to unzip?
* Python Arrays:
* Case studies

Tuple Collection

* What is tuple?
* Different ways of creating Tuple
* Method of Tuple object
* Tuple is Immutable
* Mutable and Immutable elements of Tuple
* Process tuple through Indexing and Slicing
* List v/s Tuple
* Case studies

Set Collection

* What is set?
* Different ways of creating set
* Difference between list and set
* Iteration Over Sets
* Accessing elements of set
* Python Set Methods
* Python Set Operations
* Union of sets
* functions and methods of set
* Python Frozen set
* Difference between set and frozenset ?
* Case study

Day 4

Dictionary Collection

* What is dictionary?
* Difference between list, set and dictionary
* How to create a dictionary?
* PYTHON HASHING?
* Accessing values of dictionary
* Python Dictionary Methods
* Copying dictionary
* Updating Dictionary
* Reading keys from Dictionary
* Reading values from Dictionary
* Reading items from Dictionary
* Delete Keys from the dictionary
* Sorting the Dictionary
* Python Dictionary Functions and methods
* Dictionary comprehension

Functions
* What is Function?
* Advantages of functions
* Syntax and Writing function
* Calling or Invoking function
* Classification of Functions
  * No arguments and No return values
  * With arguments and No return values
  * With arguments and With return values
  * No arguments and With return values
  * Recursion
* Python argument type functions :
  * Default argument functions
  * Required(Positional) arguments function
  * Keyword arguments function
  * Variable arguments functions
* ‘pass’ keyword in functions
* Lambda functions/Anonymous functions
  * map()
  * filter()
  * reduce()
* Nested functions
* Non local variables, global variables
* Closures
* Decorators
* Generators
* Iterators
* Monkey patching

Day 5

-
- Python Modules

* Importance of modular programming
* What is module
* Types of Modules – Pre defined, User defined.
* User defined modules creation
* Functions based modules
* Class based modules
* Connecting modules
* Import module
* From … import
* Module alias / Renaming module
* Built In properties of module

Packages
* Organizing python project into packages
* Types of packages – pre defined, user defined.
* Package v/s Folder

* py file
* Importing package
* PIP
* Introduction to PIP
* Installing PIP
* Installing Python packages
* Un installing Python packages

Exception Handling & Types of Errors

* What is Exception?
* Why exception handling?
* Syntax error v/s Runtime error
* Exception codes – AttributeError, ValueError, IndexError, TypeError…
   * Handling exception – try except block
   * Try with multi except
   * Handling multiple exceptions with single except block
* Finally block
   * Try-except-finally
   * Try with finally
   * Case study of finally block
* Raise keyword
   * Custom exceptions / User defined exceptions
   * Need to Custom exceptions
* Case studies
*