

## **Day 1**

1. Computer Programming and Python Fundamentals
  - Understand fundamental terms and definitions
  - interpreting and the interpreter, compilation and the compiler
  - lexis, syntax, and semantics
2. Understand Python's logic and structure
  - keywords
  - instructions
  - indentation
  - comments
3. Introduce literals and variables into code and use different numeral systems
  - Boolean, integers, floating-point numbers
  - scientific notation
  - strings
  - binary, octal, decimal, and hexadecimal numeral systems
  - variables
  - naming conventions
  - implementing PEP-8 recommendations
4. Choose operators and data types adequate to the problem
  - numeric operators: `** * / % // + -`
  - string operators: `* +`
  - assignment and shortcut operators
  - unary and binary operators
  - priorities and binding
  - bitwise operators: `~ & ^ | << >>`
  - Boolean operators: not, and, or
  - Boolean expressions
  - relational operators ( `= != > >= < <=` )
  - the accuracy of floating-point numbers
  - type casting
5. Perform Input/Output console operations
  - the `print()` and `input()` functions
  - the `sep=` and `end=` keyword parameters
  - the `int()` and `float()` functions

## **Day 2**

6. Control Flow – Conditional Blocks and Loops
  - conditional statements: if, if-else, if-elif, if-elif-else
  - multiple conditional statements
  - nesting conditional statements
  - building loops with while, for, range(), and in
  - iterating through sequences
  - expanding loops with while-else and for-else
  - nesting loops and conditional statements
  - controlling loop execution with break and continue
7. Data Collections – Tuples, Dictionaries, Lists, and Strings
  - constructing vectors
  - indexing and slicing
  - the len() function
  - list methods: append(), insert(), index(), etc.
  - functions: len(), sorted()
  - the del instruction
  - iterating through lists with the for loop
  - initializing loops
  - the in and not in operators
  - list comprehensions
  - copying and cloning
  - lists in lists: matrices and cubes
8. Collect and process data using tuples
  - tuples: indexing, slicing, building, immutability
  - tuples vs. lists: similarities and differences
  - lists inside tuples and tuples inside lists
9. Collect and process data using dictionaries
  - dictionaries: building, indexing, adding and removing keys
  - iterating through dictionaries and their keys and values
  - checking the existence of keys
  - methods: keys(), items(), and values()

### **Day 3**

10. Python Built-In Exceptions Hierarchy
  - BaseException
  - Exception
  - SystemExit

- KeyboardInterrupt
- abstract exceptions
- ArithmeticError
- LookupError
- IndexError
- KeyError
- TypeError
- ValueError

#### 11. Basics of Python Exception Handling

- try-except / the try-except Exception
- ordering the except branches
- propagating exceptions through function boundaries
- delegating responsibility for handling exception

### **Day 4**

#### 12. Functions

- Define and use custom functions within a Python program.
- Types of function
- Types of Arguments.
- Map Function.
- Filter Function
- Reduce Function
- Naming conventions
- Using Imports
- Documentation
- Executing Modules as Scripts
- Extended Keyword Arguments (\*args, \*\*kwargs)
- Lambda Functions
- Decorators

#### 13. Modules

- What are modules
- General Format
- Importing Modules
- Executing functions from other modules
- The \_\_name\_\_ variable

### **Day 5**

#### 14. Python Dates and Times

- Understanding Time

- The Time Module
- The datetime Module

#### 15. Math

- Airthmetic Operators
- Assignment Operators
- Built-in Math Functions
- The math Module
- The random Module
- Module 8: File Processing
- Opening Files
- The os and os.path modules
- Reading files
- Writing into a file
- Appending data into a file

### **Day 6**

#### 16. OOPS in Python

- Introduction to Object-Oriented Python
- Creating Classes, Methods, and Objects
- Using Constructor and Attributes
- Using Class Attributes and Static Methods
- Understanding Object Encapsulation
- Private Attributes and Methods
- Controlling Attribute Access
- Creating and Accessing Properties

### **Day 7**

#### 17. Playing with Data

- Relational Databases
- CSV
- Getting Data from the Web
- JSON
- Overview of Data Serialization
- Importance of Data Interchange Formats
- Understanding JSON syntax and structure
- Encoding Python Data to JSON - Serialization
- Decoding JSON to Python Data – Deserialization
- Introduction to XML
- Parsing XML with xml.etree.ElementTree

- Best Practices in XML Processing
- Introduction to CSV Format
- Reading CSV data
- Writing Data into CSV files
- Best Practices for CSV Processing
- Integrating JSON, XML and CSV using advanced python library Pandas

## **Day 8**

### 18. NumPy

- Introduction to NumPy
- Installation
- NumPy Arrays (numpy.array)
- Array Indexing and Slicing
- Array Shape and Reshaping
- Array Operations (Sum, Mean, etc.)
- numpy.arange and numpy.linspace
- Array Stacking and Splitting
- Data Type Promotion
- Formation of 1D, 2D and 3D arrays
- Exploring dimension type, shape and size of array
- Exploring dimension type, shape and size of array
- Indexing of NumPy Arrays
- Slicing of NumPy Arrays
- One's Matrix and Zero's Matrix
- Identity Matrix
- min() and argmin()
- max() and argmax()
- np.sort()
- np.argsort()
- Addition of matrices
- Multiplication of matrices

## **Day 9**

### 19. Pandas

- Introduction to Pandas
- Series
- DataFrames
- Missing Data
- Merging Joining and Concatenating
- Operations
- Data Input and Output

- Groupby()
- Pivoting
- VLookup