

DAY 1:

Introduction to Python:

- a) History and features of Python programming language
- b) Setting up the development environment
- c) Running Python scripts and interactive mode

DAY 2:

Variables and Data Types:

- a) Understanding variables and their usage
- b) Numeric data types (integers, floats, etc.)
- c) Strings and string manipulation
- d) Boolean data type and logical operations

DAY 3:

Control Structures:

- a) Conditional statements (if, elif, else)
- b) Looping structures (for loops, while loops)
- c) Control flow (break, continue)

DAY 4:

Lists and Tuples:

- a) Creating and manipulating lists
- b) List comprehension
- c) Understanding tuples and their immutability

DAY 5:

Dictionaries and Sets:

- a) Working with dictionaries (key-value pairs)
- b) Dictionary manipulation and iteration
- c) Sets and their operations (union, intersection, etc.)

DAY 6:

Functions and Modules:

- a) Defining and calling functions
- b) Function parameters and return values
- c) Modular programming using modules and importing

DAY 7:

File Handling:

- a) Reading from and writing to files
- b) File modes and operations (read, write, append)
- c) Handling file exceptions

DAY 8:

Exception Handling:

- a) Understanding exceptions and their types
- b) Using try-except blocks to handle exceptions
- c) Handling multiple exceptions and custom exceptions

DAY 9:

Object-Oriented Programming (OOP) Basics:

- a) Introduction to OOP concepts (classes, objects, methods)
- b) Encapsulation, inheritance, and polymorphism
- c) Class attributes and instance attributes

DAY 10:

OOP Advanced Concepts:

- a) Class inheritance and method overriding
- b) Abstract classes and interfaces
- c) Class composition and aggregation

DAY 11:

Recursion:

- a) Understanding recursion and recursive functions
- b) Recursive algorithms and problem-solving techniques

DAY 12:

String Manipulation:

- a) String methods and operations
- b) String formatting and interpolation
- c) Regular expressions for advanced string manipulation

DAY 13:

Sorting and Searching Algorithms:

- a) Sorting algorithms (e.g., bubble sort, merge sort)
- b) Searching algorithms (e.g., linear search, binary search)
- c) Analyzing algorithm complexity (Big O notation)

DAY 14:

Data Structures - Stack and Queue, Linked List, Trees, Graphs:

- a) Understanding and implementing stack and queue data structures
- b) Linked list operations (insertion, deletion, traversal)
- c) Introduction to tree and graph data structures

DAY 15:

File Handling and Data Serialization:

a) Working with files and directories b) Serializing data using pickle and JSON

DAY 16:

Regular Expressions:

- a) Introduction to regular expressions and their syntax
- b) Pattern matching and substitution using regular expressions

DAY 17:

Debugging and Testing:

- a) Debugging techniques and tools
- b) Unit testing using the unittest module

DAY 18:

Mini Projects:

- a) Implementing small projects to apply learned concepts b) Enhancing problem-solving and coding skills