Python Syllabus

Starting off with Python

- Features and Advantages of Python
- Working principle of Python interpreter
- Installation of Python
- Installation of PyCharm IDE
- Installation of Anaconda

Python Virtual environment

- Use of virtual environment
- Steps to create virtual environment
- Activate / Deactivate virtual environment
- Creation of New Virtual environment using PyCharm IDE

First Python Program

- Run Python Program
- Assign value to python variables
    - Method-1: Direct Initialisation
    - Method 2: using conditional Operator (?:)
- Assign Multiple values to Multiple variables
- Local variables and Global variables
    - Memory allocation
    - Local Variable
    - Global Variable
    - Local Variable (vs) Global Variables
    - Difference between Local and Global variable
    - Advantages
    - Disadvantages

PEP8 Standards

- What is PEP8?
- PEP8 Guidelines

- - Naming Convention
    - Indentation
      - Top-level functions and classes with two lines (2 blank lines)
      - Single blank line inside the function
    - Docstring
    - Importing module
    - Block Comment
    - Inline comments
    - Rules to follow
  - What are Programming Errors?
  - Types of Programming Errors
    - Syntax Errors
    - Run time Errors Logical Errors (Exceptions)

String formatting

- % formatting
- str.format()
- F-strings
- Multiline f-strings
- Speed comparison
- Sep and end

Python Fundamentals

- Python Keywords
- Python Identifiers

- Type casting or Type conversion
  - Implicit Conversion
  - Explicit type casting
- Isinstance() Method
- ID() function
- help() function
- dir() function
- len() function
- range()
- enumerate()
- eval() method
- Input Function
- Copy
  - Shallow copy
  - Deep copy

Python Operators
- Arithmetic Operators
- Comparison Operators
- Logical Operators
- Identity Operators
- Membership Operators

Program Control Flow
- Conditional Statements
  - If condition
  - If – else condition
  - if-elif-else condition
- Loops
  - Finite Loops
    - For loop
      - Looping with numbers
      - Looping with lists
  - Infinite Loops
    - While Loop
      - While loop with numbers
      - While loop with lists
- Loop Control Statements

- o break
- o continue
- o pass
- o for else
- Coding Exercises

Strings

- What is String?
- Define a String
- Creating a String
- Access characters in String
  - o Positive Index Slicing
  - o Negative Index Slicing
  - o Slice with Stride
- Delete String
- Python String Operations
  - o Concatenate literal strings
  - o Concatenate strings using + operator
  - o Concatenate strings using += operator
  - o Concatenate strings using the join() method
  - o Concatenate strings using the %-formatting
  - o Concatenate strings using the format() method
  - o Concatenate strings using f-strings
- String methods
  - o lower ()
  - o upper ()
  - o strip ()
  - o replace ()
  - o split ()
  - o join ()
  - o find ()
  - o index ()
  - o isalnum ()
  - o isalpha ()
  - o isdigit ()
  - o startswith ()
  - o endswith ()
- Coding exercises

List

- What is a List?
- Define a List
- Creating a List
- Access the List elements
    - Negative Indexing
- Adding List Elements
    - append () method
    - insert () method
    - extend () method
- Remove the List elements
    - using remove () method
    - using pop () method
    - using clear () method
- List Methods
    - index ()
    - count ()
    - sort ()
    - copy ()
- Slicing of a List
- List comprehension
- Zip
- Multiple ways to iterate over list elements
    - for loop
    - for loop and range()
    - while loop
- Coding exercises

Tuples

- What is Tuple?
- Define a Tuple
- Creating a Tuple
- Accessing of Tuple elements
    - Positive Index
    - Negative Index
- Concatenation of Tuples
    - Method #1: using + operator

- o   Method #2: using sum()
- Delete a Tuple
- Tuple built-in methods
  - o   index ()
  - o   count ()
- Coding Exercises

Set

- What is Set?
- Define a Set
- Creating a Set
- Adding Elements to Set
  - o   add () method
  - o   update () method
- Accessing a Set
- Remove set elements
  - o   discard () method
  - o   pop () method
  - o   clear () method
- Frozen sets
- Set Methods
  - o   union ()
  - o   union using the | operator
  - o   difference ()
  - o   intersection ()
  - o   intersection using & operator
  - o   issuperset () & issubset()
- Coding Exercises

Dictionary

- What is a Dictionary?
- Create a Dictionary
- Add Elements
  - o   Change Value
- Update dictionary
- Access Items
  - o   Get keys
  - o   Get Values

- o   Get Items
- o   Check if Key Exists
  - ▪   use in to see if a key exists
  - ▪   use .keys()
  - ▪   use dictionary[key]
  - ▪   use the .get Method
- ●   Remove elements
  - o   del keyword
  - o   pop () method
  - o   popitem () method
  - o   clear ()
- ●   Loop Dictionaries
  - o   using comprehensions
  - o   Turning Keys into Values and Vice Versa
  - o   Filtering Items
  - o   Removing Specific Items
  - o   Sorting a Dictionary
  - o   Sorted by Keys
  - o   Sorted by Values
  - o   Iterating with .popitem()
- ●   Copy a Dictionary
  - o   copy ()
  - o   dict ()
- ●   Merge more than one dictionary
  - o   Using the | Operators
  - o   Using the ** Operator
- ●   len()
- ●   fromkeys()
- ●   setdefault()
- ●   Coding Exercises

Function

- ●   Define a Function
- ●   Creating a Function
- ●   Passing Function arguments
- ●   Calling a Function
- ●   Arguments
  - o   Arbitrary arguments (*args)

- o   Positional arguments
- o   Keyword arguments
- o   Default arguments
- o   Arbitrary Keyword arguments (**kwargs)
- Recursive Function (a function that calls itself)
- Coding Exercises

Modules & Packages

- Module
- Package

map () Function

- what is map() Function
- Coding Exercises

filter () Function

- what is filter() Function
- Coding Exercises

reduce () Function

- what is reduce Function
- Coding Exercises

Lambda Function

- Lambda Function
- Why use Lambda Function?
- Lambda with map ()
- Lambda with filter ()
- Lambda with reduce ()

Iterators

- What is Python Iterator?
- Creation of Python Iterator
- __iter__(iterable)
- next (__next__)
- implicit creation of iterators
- For -loop and Iterators

Generators

- What is a Python Generator?
- Define a Generator- Function
- Generator – object
- Yield keyword

Decorators

- Assigning Function to Variables
- Defining Functions inside other Functions
- Passing Functions as Arguments to other Functions
- Functions returning other Functions
- What is Decorator?
- Creating a Decorator
- Applying Multiple Decorators to a Single Function
- Accepting Arguments in a Decorator Functions
- Debugging Decorators

Object Oriented Programming

- attributes
- behavior
- Class
- Object
- Constructors
- Methods
  - class methods
  - static methods
  - method overloading
  - method overriding
- Encapsulation
- Inheritance
- Types of Inheritance
  - Single Inheritance
  - Multiple Inheritance
  - Multilevel Inheritance
  - Hierarchical Inheritance
- Polymorphism
- Abstraction

Exception Handling

- What is an Exception?
- What is Error?
- Difference between Errors and Exceptions
- Catching Exceptions
  - Try and Except Statement
  - Finally, Keyword
- Raising Exception

File Handling

- Files
- Open () Function
- Access Modes
- read () mode
- closing Files
- write () mode
- with clause
- File methods
  - readline ()
  - readlines ()
  - writeline ()
  - writelines ()
  - seek ()
  - flush ()

JSON

- What is JSON
- Converting from JSON to Python
  - Parsing JSON
    - loads ()
  - Return JSON Object
    - load ()
- Reading JSON file
- Convert from Python to Json
  - Converting to Json String
    - dumps ()
  - Writing to a JSON file
    - dump ()