

# Core Python

## Introduction to Python

- Origin of Python
- Introduction to Python
- Applications of Python
- Features and Versions of Python
- Languages used in Python Development

## Python Basics

- Interactive Mode vs. Script Mode
- Interpreter vs. Compiler
- Scripting vs. Programming Languages
- Why Learn Python?
- Python Indentation
- Comments and Quotations
- Identifiers and Keywords
- Variables:
  - Assigning values in different ways
  - Print(), type(), and id()
  
- User Input Handling
- Working with Input Functions
- Python Data Types
- Type Conversions and eval()

# Assignment - 1

## Data Structures in Python

- Introduction to Data Structures

## String Data Structure

- Definition and Creation Methods
- String Indexing & Slicing
- Concatenation & Multiplication
- String Methods (capitalize, title, count, replace, upper, lower, etc.)
- String Immutability
- String Sorting & Reversal
- String Length & Searching Methods
- String Validations (isdigit(), isalpha(), isalnum(), etc.)

## List Data Structure

- Creating Lists (Homogeneous & Heterogeneous)
- Indexing & Slicing
- List Operations (Concatenation, Multiplication, Unpacking, Sorting, etc.)
- Range and xrange Functions
- List Methods (insert, append, pop, remove, clear, etc.)

## Tuple Data Structure

- Creating & Working with Tuples
- Tuple Indexing & Slicing
- Tuple Operations (Concatenation, Multiplication, Sorting, etc.)
- Tuple Immutability & Nested Tuples

## Set Data Structure

- Creating & Modifying Sets
- Normal Sets vs. Frozen Sets
- Set Operations (Union, Intersection, Difference, etc.)
- Set Conversion Methods

## Dictionary Data Structure

- Dictionary Creation & Operations
- Key-Value Pair Management
- Dictionary Methods (keys(), values(), pop(), popitem(), etc.)
- Conversion between Data Structures

## Assignment - 2

## Operators in Python

- Arithmetic, Logical, Assignment, Comparison, Bitwise, Identity, Membership Operators

# Functions & Arguments

- Defining & Calling Functions
- Named, Keyword, Default & Positional Arguments
- `*args` and **kwargs``
- Nested Functions & Variable-Length Arguments

# Lambda Functions

- Lambda Expressions
- Filter, Map, and Reduce Functions

# Control Statements & Looping

- If, If-Else, Elif Statements
- Membership Testing for Various Data Structures
- For & While Loops
- Break, Continue, and Pass Statements
- Iterating Over Data Structures

# Assignment - 3

# Advanced Python

# File Handling

- File Creation, Opening, Reading, Writing, and Appending
- Modes of Operations
- Seek & Tell Methods
- File Handling Real-Time Scenarios

## Object-Oriented Programming (OOPs)

- Class & Object
- Instance & Class Variables
- Constructor & Data Hiding
- Method Overloading & Overriding
- Abstraction, Inheritance, Polymorphism, Encapsulation

## Modules & Exception Handling

- Python Modules & Importing Methods
- Standard & User-Defined Modules
- Exception Handling (try-except, finally, assert)

## Logging in Python

- Creating and Storing Logs
- Log Levels (DEBUG, INFO, WARNING, ERROR, CRITICAL)

## Iterators, Generators & Decorators

- Yield Keyword
- Difference between Yield & Return
- Function Decoration

## Unit Testing in Python

- `unittest` Module
- Assert Functions

## Regular Expressions

- Regex Basics & Methods (findall, search, match, etc.)
- Web Scraping Techniques

## Command Line Arguments & Database Handling

- Reading Command Line Inputs
- Connecting Python with Databases
- Executing SQL Commands in Python
- Fetching & Updating Data

## End of Course

- Real-time Case Studies
- Project Implementation
- Interview Preparation