PYTHON

Course Brochure

PYTHON COURSE CONTENT

1) Language Fundamentals
2) Operators
3) Flow Control
4) String Data Type
5) List Data Structure
6) Tuple Data Structure
7) Set Data Structure
8) Dictionary Data Structure
9) Functions
10) Modules
11) Packages
12) 100 Pattern Programs
13) OOP's Part – 1
14) OOP's Part – 2
15) OOP's Part – 3
16) OOP's Part – 4
17) Exception Handling
18) File Handling
19) Multi Threading
20) Python Database Programming
21) Regular Expressions & Web Scraping
22) Decorator Functions
23) Generator Functions
24) Assertions
25) Python Logging

DETAILED CONTENT

2) Float Data Type
3) Complex Data Type
4) bool Data Type
5) str Data Type
6) bytes Data Type
7) bytearray Data Type
8) List Data Type
9) Tuple Data Type
10) Range Data Type
11) Set Data Type
12) frozenset Data Type
13) dict Data Type
14) None Data Type

- ❖ Base Conversions
- ❖ Slicing of Strings

- ❖ TYPE CASTING
    - int()
    - float()
    - complex()
    - bool()
    - str()

- ❖ Fundamental Data Types vs Immutability
- ❖ Escape Characters
- ❖ Constants

2) OPERATORS

1) Arithmetic Operators
2) Relational Operators OR Comparison Operators
3) Equality Operators
4) Logical Operators

5) Bitwise Oeprators
    - Bitwise Complement Operator (~)

6) Shift Operators
    - << Left Shift Operator
    - >> Right Shift Operator

7) Assignment operators
8) Ternary Operator OR Conditional Operator

9) Special operators
    - Identity Operators
    - Membership operators

- Operator Precedence
- Mathematical Functions (math Module)
- Command Line Arguments
- Output Statements

3) FLOW CONTROL

- Conditional Statements
    - if
    - if-elif
    - if-elif-else

- Iterative Statements
    - for
    - while

- Transfer Statements
    - break
    - continue
    - pass

- Loops with else Block
- del Statement
- Difference between del and None

4) STRING DATA TYPE

- What is String?
- How to define multi-line String Literals?

- How to Access Characters of a String?
    - Accessing Characters By using Index
    - Accessing Characters by using Slice Operator

- Behaviour of Slice Operator
- Slice Operator Case Study
- Mathematical Operators for String
- len() in-built Function
- Checking Membership
- Comparison of Strings

- Removing Spaces from the String
  - rstrip()
  - lstrip()
  - strip()

- Finding Substrings
- Counting substring in the given String
- Replacing a String with another String
- Splitting of Strings
- Joining of Strings
- Changing Case of a String

- Checking Starting and Ending Part of the String
  - s.startswith(substring)
  - s.endswith(substring)

- To Check Type of Characters Present in a String
- Formatting the Strings

- Important Programs regarding String Concept
  1) Program to Reverse the given String
  2) Program to Reverse Order of Words
  3) Program to Reverse Internal Content of each Word
  4) Program to Print Characters at Odd Position and Even Position for the given String
  5) Program to Merge Characters of 2 Strings into a Single String by taking Characters alternatively
  6) Program to Sort the Characters of the String and First Alphabet Symbols followed by Numeric Values
  7) Program for the following Requirement
     (Input: a4b3c2, Output: aaaabbbcc)

8) Program to perform the following Activity
   (Input: a4k3b2, Outpt: aeknbd)
9) Program to Remove Duplicate Characters from the given Input String
10) Program to find the Number of Occurrences of each Character present in the given String
11) Program to perform the following Task
    - Input: 'one two three four five six seven'
    - Output: 'one owt three ruof five xis seven'

- Formatting the Strings

5) LIST DATA STRUCTURE

- Creation of List Objects
- Accessing Elements of List
  - By using Index
  - By using Slice Operator

- List vs Mutability

- Traversing the Elements of List
  - By using while Loop
  - By using for Loop
  - To display only Even Numbers
  - To display Elements by Index wise

- Important Functions of List
  - To get Information about List
    - len()
    - count()
    - index()

  - Manipulating Elements of List
    - append()
    - insert()
    - extend()
    - remove()
    - pop()

- Ordering Elements of List
    - reverse()
    - sort()

- Using Mathematical Operators for List Objects
    - Concatenation Operator (+)
    - Repetition Operator (*)

- Comparing List Objects

- Membership Operators
    - in Operator
    - not in Operator

- clear() Function
- Nested Lists
- Nested List as Matrix
- List Comprehensions

6) TUPLE DATA STRUCTURE

- Tuple Creation

- Accessing Elements of Tuple
    - By using Index
    - By using Slice Operator

- Tuple vs Immutability

- Mathematical Operators for Tuple
    - Concatenation Operator (+)
    - Multiplication Operator OR Repetition Operator (*)

- Important Functions of Tuple
    - len()
    - count()
    - index()
    - sorted()
    - min() And max()
    - cmp()

- Tuple Packing and Unpacking
- Tuple Comprehension
- Differences between List and Tuple

7) SET DATA STRUCTURE

- Creation of Set Objects
- Important Functions of Set
    - add(x)
    - update(x,y,z)
    - copy()
    - pop()
    - remove(x)
    - discard(x)
    - clear()
- Mathematical Operations on the Set
    - union()
    - intersection()
    - difference()
    - symmetric_difference()

- Membership Operators: (in, not in)
- Set Comprehension

8) DICTIONARY DATA STRUCTURE

- How to Create Dictionary?
- How to Access Data from the Dictionary?
- How to Update Dictionaries?

- How to Delete Elements from Dictionary?
    - del d[key]
    - d.clear()
    - del d

- Important Functions of Dictionary
    - dict()
    - len()
    - clear()
    - get()

- pop()
- popitem()
- keys()
- values()
- items()
- copy()
- setdefault()
- update()

- Dictionary Comprehension

9) FUNCTIONS

- Built in Functions
- User Defined Functions
- Parameters
- Return Statement
- Returning Multiple Values from a Function

- Types of Arguments
  - Positional Arguments
  - Keyword Arguments
  - Default Arguments
  - Variable Length Arguments

- Case Study

- Types of Variables
  - Global Variables
  - Local Variables

- global Keyword
- Recursive Functions
- Anonymous Functions
- Normal Function
- Lambda Function
- filter() Function
- map() Function
- reduce() Function

- Everything is an Object
- Function Aliasing
- Nested Functions

10) MODULES

- Renaming a Module at the time of import (Module Aliasing)
- from ... import
- Various Possibilties of import
- Member Aliasing
- Reloading a Module
- Finding Members of Module by using dir() Function
- The Special Variable __name__
- Working with math Module
- Working with random Module
    - random() Function
    - randint() Function
    - uniform() Function
    - randrange ([start], stop, [step])
    - choice() Function

11) PACKAGES

12) 100 PATTERN PROGRAMS

13) OOP's Part – 1

- What is Class?
- How to define a Class?
- What is Object?
- What is Reference Variable?
- Self Variable
- Constructor Concept
- Differences between Methods and Constructors

- Types of Variables
    - Instance Variables (Object Level Variables)
    - Static Variables (Class Level Variables)
    - Local variables (Method Level Variables)

- Where we can declare Instance Variables
    - Inside Constructor by using self variable
    - Inside Instance Method by using self variable
    - Outside of the class by using object reference variable

- How to Access Instance Variables
- How to delete Instance Variable from the Object

- Static Variables
- Instance Variable vs Static Variable
- Various Places to declare Static Variables
- How to access Static Variables
- Where we can modify the Value of Static Variable
- How to Delete Static Variables of a Class
- Local Variables

- Types of Methods
    - Instance Methods
    - Class Methods
    - Static Methods

- Setter and Getter Methods
- Passing Members of One Class to Another Class
- Inner Classes
- Garbage Collection
- How to enable and disable Garbage Collector in our Program
- Destructors
- How to find the Number of References of an Object

14) OOP's Part – 2

- Inheritance
    - By Composition (Has-A Relationship)
    - By Inheritance (IS-A Relationship

- IS-A vs HAS-A Relationship
- Composition vs Aggregation

- Types of Inheritance
    - Single Inheritance
    - Multi Level Inheritance
    - Hierarchical Inheritance
    - Multiple Inheritance
    - Hybrid Inheritance
    - Cyclic Inheritance

- Method Resolution Order (MRO)
- Head Element vs Tail Terminology
- How to find Merge?
- Finding mro(P) by using C3 Algorithm
- super() Method
- How to Call Method of a Particular Super Class?
- Various Important Points about super()

15) OOP's Part – 3

- Polymorphism
- Duck Typing Philosophy of Python

- Overloading
    - Operator Overloading
    - Method Overloading
    - Constructor Overloading

- Overriding
    - Method Overriding
    - Constructor Overriding

16) OOP's Part – 4

- Abstract Method
- Abstract class
- Interface
- Concreate Class vs Abstract Class vs Inteface
- Public,Private and Protected Members
- __str__() Method
- Difference between str() and repr() Functions

- Small Banking Application

17) Exception Handling

- Syntax Errors
- Runtime Errors
- What is Exception
- Default Exception Handing in Python
- Python's Exception Hierarchy
- Customized Exception Handling by using try-except
- Control Flow in try-except
- How to Print Exception Information
- try with Multiple except Blocks
- Single except Block that can handle Multiple Exceptions
- Default except Block
- finally Block
- Control Flow in try-except-finally
- Nested try-except-finally Blocks
- Control Flow in nested try-except-finally
- else Block with try-except-finally
- Various possible Combinations of try-except-else-finally

- Types of Exceptions
  - Predefined Exceptions
  - User Definded Exceptions

- How to Define and Raise Customized Exceptions

18) File Handling

- Types of Files
  - Text Files
  - Binary Files

- Opening a File
- Closing a File
- Various Properties of File Object

- Writing Data to Text Files

- write(str)
- writelines(list of lines)

- Reading Character Data from Text Files
  - read() → To Read Total Data from the File
  - read(n) → To Read 'n' Characters from the File
  - readline() → To Read only one Line
  - readlines() → To Read all Lines into a List

- The with Statement
- The seek() and tell() Methods
- How to check a particular File exists OR not
- Handling Binary Data
- Handling CSV Files
- Writing Data to CSV File
- Reading Data from CSV File
- Zipping and Unzipping Files
- To Create Zip File
- Working with Directories
- Running Other Programs from Python Program
- How to get Information about a File
- Pickling and Unpickling of Objects

19) Multi Threading

- Multi Tasking
  - Process based Multi Tasking
  - Thread based Multi Tasking

- The ways of Creating Thread in Python
  - Creating a Thread without using any class
  - Creating a Thread by extending Thread class
  - Creating a Thread without extending Thread class

- Setting and Getting Name of a Thread
- Thread Identification Number (ident)
- enumerate() Function
- isAlive() Method

- join() Method
- Daemon Threads
- Default Nature

- Synchronization
  - Lock
  - RLock
  - Semaphore

- Synchronization By using Lock Concept
- Problem with Simple Lock
- Demo Program for Synchronization by using RLock
- Difference between Lock and RLock
- Synchronization by using Semaphore
- Bounded Semaphore
- Difference between Lock and Semaphore
- Inter Thread Communication
- Inter Thread Communication by using Event Objects

- Methods of Event Class
  - set()
  - clear()
  - isSet()
  - wait()|wait(seconds)

- Inter Thread Communication by using Condition Object

- Methods of Condition
  - acquire()
  - release()
  - wait()|wait(time)
  - notify()
  - notifyAll()

- Case Study
- Inter Tread Communication by using Queue

- Important Methods of Queue
    - put()
    - get()

- Types of Queues
    - FIFO Queue
    - LIFO Queue
    - Priority Queue

- Good Programming Practices with usage of Locks

20) Python Database Programming

- Storage Areas
    - Temporary Storage Areas
    - Permanent Storage Areas
- File Systems
- Databases
- Python Database Programming
- Working with Oracle Database
- Installing cx_Oracle
- How to Test Installation
- Working with MySQL Database
- Commonly used Commands in MySQL
- Driver/Connector Information
- How to Check Installation

21) Regular Expressions & Web Scraping

- Character Classes
- Pre defined Character Classes
- Qunatifiers
- Important Functions of Remodule
    1) match()
    2) fullmatch()
    3) search()
    4) findall()
    5) finditer()
    6) sub()

7) subn()
8) split()
9) compile()

- Web Scraping by using Regular Expressions

22) Decorator Functions

Decorator Chaining

23) Generator Functions

- Advantages of Generator Functions
- Generators vs Normal Collections wrt Performance
- Generators vs Normal Collections wrt Memory Utilization

24) Assertions

- Debugging Python Program by using assert Keyword
- Types of assert Statements
    - Simple Version
    - Augmented Version
- Exception Handling vs Assertions

25) Python Logging

- Logging Levels
- How to implement Logging
- How to configure Log File in over writing Mode
- How to Format Log Messages
- How to add Timestamp in the Log Messages
- How to Change Date and Time Format
- How to write Python Program Exceptions to the Log File
- Problems with Root Logger
- Need of Our Own Customized Logger
- Advanced logging Module Features: Logger
- Logger with Configuration File
- Creation of Custom Logger
- How to Create seperate Log File based on Caller
- Advantages of Customized Logger