TOPSTECHNOLOGIES
Training | Outsourcing | Placement | Study Abroad

Python Lectureflow

Module-1) SE - Overview of IT Industry	5
• Introduction of students • Career in IT • Understanding Student Login of TOPS ERP • Using Lab • What is Program • What is programming? • Types of Programming Language • World Wide Web • How Internet Works • Network Layers on Client and Server • Client And Servers • Types of Internet Connections • Protocols • Application Security • Software Applications and its types • Software Architecture • Layers in Software Architecture • Software Environments • Types of Programming Languages • Source Code • Github and introductions • Student Account in Github • Types of Software • Introduction of Software • Application software • Software development process • Software Requirement • Software Analysis • System Design • Software Testing • Maintenance • Development • Web Application • Designing • moble application • DFD • Desktop Application • Flow Chart	

Module 3) SE - Introduction to Programming - May 2024	**15**

- Overview of C Programming - What is C Programming? History and Evolution, Importance and Applications
- Setting Up Environment -Installing a C Compiler (e.g., GCC), Choosing an IDE (DevC++. VS Code, Codeblocks, etc) Writing Your First Program
- Basic Structure of a C Program - Structure of a C Program ,Comments in C, Data Types and Variables, Constants, Keywords and Identifiers
- Operators - Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Bitwise Operators, Conditional Operator
- Control Flow Statements - Decision-Making in C - If Statement, If_Else statement, If_elseif (Elseif Ladder), Nested if-else statement, Switch statement
- Looping in C - While Loop, For Loop, Do-While Loop
- Loop Control Statements - Break, Continue, Go to
- Functions in C - Introduction to Functions, Function Declaration and Definition, Function Call and Return
- Arrays in C - Introduction to Arrays, One-Dimensional Arrays, Multi-Dimensional Arrays
- Pointers in C - Introduction to Pointers, Pointer Declaration and Initialization
- Strings in C - Introduction to Strings, String Handling Functions, strlen, strcpy, strcat, strcmp, strcmpi,strchr, String Input and Output
- Structures in C - Introduction to Structures, Structure Declaration and Initialization, Array of Structure Nested Structures
- File Handling in C, File Operations (Opening, Closing, Reading, and Writing), File Pointers, File Handling Functions

Module-4) Introduction to OOPS Programming	**8**

- Introduction to C++ - Understanding the Basics of Programming, Introduction to C++ Language, POP Vs OOP, Advantages of OOP, Setting Up C++ Development Environment, Writing and Running Your First C++ Program, Input and Output Operations in C++
- Variables, Data Types, and Operators - Variables and Constants in C++, Data Types and Size Specifiers, Assignments, Arithmetic, Relational, Logical, and Bitwise Operators, Type Conversion in C++, Constants and Literals
- Control Flow Statements - Conditional Statements: if, if_else, else if ladder, nested if, Switch Statement, Loops: while, do-while, for, Break and Continue Statements, Nested Control Structures
- Functions and Scope - Introduction to Functions ,Function Prototypes and Definitions, Parameters and Return Values, Scope of Variables
- Arrays and Strings - Introduction to Arrays, Single-Dimensional and Multi-Dimensional Arrays, Array Initialization, Accessing Elements, and Manipulation, Introduction to Strings in C++, String Operations and Functions
- Introduction to Object-Oriented Programming -Understanding the Basics of Object-Oriented Programming, Advantages of OOP Paradigm, Key Concepts: Classes, Objects, Inheritance, Polymorphism, Encapsulation, Introduction to C++ as an Object-Oriented Language

- Classes and Objects - Declaring Classes and Objects in C++, Class Members: Data Members and Member Functions, Constructors and Destructors, Access Specifiers: Public, Private, Protected, Class Member Functions: Inline and Outside Definitions
- Inheritance - Concept of Inheritance and Reusability, Types of Inheritance: Single, Multiple, Multi-level, Hierarchical, Base and Derived Classes in C++, Access Control and Inheritance, Constructors and Destructors in Inheritance
- Polymorphism - Understanding Polymorphism in OOP, Compile-Time and Runtime Polymorphism, Function Overloading and Operator Overloading, Virtual Functions and Dynamic Binding, Abstract Classes and Pure Virtual Functions, Scope resolution operator, Static Keywords, Inline Function
- Encapsulation - Understanding Encapsulation and Information Hiding, Access Specifiers: Public, Private, Protected, Encapsulation in C++ Classes, Benefits of Encapsulation in OOP, Friend Functions and Friend Classes
- File Handling - Introduction to File Handling in C++, Opening, Closing, Reading, and Writing Files, Error Handling in File Operations, Working with File Streams

Module-5) SE - Introduction to DBMS	**9**

- Introduction to SQL - SQL Overview, Definition of SQL, Importance of SQL in Database Management, DBMS-RDBMS
- SQL Syntax - Basic SQL Syntax, Structure of SQL Statements
- SQL Constraints - Types of Constraints (NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY), Implementing Constraints in Tables
- Main SQL Commands and Sub-commands - Data Definition Language (DDL), CREATE Command, Creating Tables, Specifying Column Names, Data Types, and Constraints
- ALTER Command - Modifying Existing Tables, Adding, Modifying, or Dropping Columns
- DROP Command - Deleting Tables from the Database
- Data Manipulation Language (DML) - INSERT Command, Adding Data into Tables Specifying Column Names and Values
- UPDATE Command - Modifying Existing Data in Tables, Changing Values in Specific Columns
- DELETE Command - Removing Data from Tables, Deleting Specific Rows with WHERE Clause
- Data Query Language (DQL) - SELECT Command, Retrieving Data from Tables, Filtering Data with WHERE Clause, Sorting Data with ORDER BY Clause, Limiting Results with LIMIT or FETCH FIRST Clause
- Data Control Language (DCL) -GRANT Command, Granting Privileges to Users or Roles, Granting SELECT, INSERT, UPDATE, DELETE Permissions
- REVOKE Command - Revoking Privileges from Users or Roles, Removing SELECT, INSERT, UPDATE, DELETE Permissions
- Transaction Control Language (TCL) - COMMIT Command, Saving Changes Permanently to the Database
- ROLLBACK Command - Reverting Uncommitted Changes, ?SAVEPOINT Command - Creating Intermediate Points in a Transaction for Rollback
- SQL Joins - Inner Join, Left Join, Right Join, Full Outer Join
- SQL Group By - Grouping Data in SQL Queries

- SQL Stored Procedure - Definition and Purpose of Stored Procedures, Creating and Executing Stored Procedures
- SQL View - Creating Views in SQL, Advantages of Using Views
- SQL Trigger - Introduction to Triggers, Types of Triggers (INSERT, UPDATE, DELETE)
- Introduction to PL/SQL - Definition and Purpose of PL/SQL, Benefits of Using PL/SQL
- PL/SQL Syntax - Structure of PL/SQL Blocks, Variables, Constants, and Data Types in PL/SQL
- PL/SQL Control Structures - IF-THEN, IF-THEN-ELSE, CASE Statements ,Loops (WHILE, FOR)
- SQL Cursor - Introduction to Cursors in PL/SQL, Implicit vs. Explicit Cursors
- Rollback and Commit Savepoint - Transaction Management in PL/SQL

Module 13) Python - Fundamentals of python language	8

- Introduction of students
- Understanding Student Login of TOPSERP
- Career in IT
- Using Lab
- Introduction of Python
- Programming Style
- Core python concepts
- Conditional Statements
- If- else Nested if-else
- Practical Examples: 1) How to the python code Structure work? 2) How to create variable in python? 3) How to take user input? 4) How to check the type of variable dynamically. 5) W.A.P to find greater and less than number using If_else 6) W.A.P to find prime number using if_else 7) W.A.P to find the grade according to percentage using if_else ladder. 8) W.A.P to find that who can donate the blood using Nested if.
- Looping For , While
- Generators and Iterators
- Nested loops
- map(), reduce(), filter() and Closures and Decorators
- Control Statements
- 1) WAP to print each fruit in list using simple for loop. List1 (apple,banana,mango) 2) WAP to find the length of string using simple for loop List1 (apple,banana,mango) 3) WAP to find particular string using simple for loop and simple if condition. 4) Print this pattern using nested for Loop.
- Break
- Continue
- Pass
- Practical Example: 1) W.A.P to skip the (Banana) from the list using Continue Statement List1 - (apple,banana,mango) 2) W.A.P to break the for loop when (Banana) get in if Condition.
- String Manipulation
- Accessing Strings
- Basic Operations
- String slices
- Function and Methods

- 1) W.A.P to print (Hello) using string 2) W.A.P to allocate the string to a variable. 3) W.A.P to print String using three quotes 4) W.A.P to access the 1st position character using index value. 5) W.A.P to Access the string after the index value 1. 6) W.A.P to Access the string before the index value 5. 7) W.A.P to Access the String between the index value 1 to 4 8) W.A.P to print the string from the last index value. 9) W.A.P to print the String alternate character after the index value 1. 10) W.

Module 14) Python - Collections, functions and Modules in Python	**12**

- Accessing list
- Operations
- Working with List
- Function and Method
- Practical Example: 1) W.A.P create the list of multiple datatype element. 2) W.A.P to find the length of the list. 3) W.A.P to update the list using the insert() and append() 4) W.A.P to remove the element using the pop() and remove()
- Tuple
- Accessing Tuples
- Operations Working
- Functions and Method
- Dictionaries
- Accessing value in dictionaries
- Working with dictionaries
- Property
- Practical Example: 1) W.A.P to access value on index value in the list 2) W.A.P to access the value after the index value 1. 3) W.A.P to access the value between 1 to 5 4) W.A.P to access the value till index 5. 5) W.A.P to update the list using the index value. 6) W.A.P to irate the list using for loop. 7) W.A.P to insert the value in empty list using for loop and append(). 8) W.A.P to delete the element using del() 9) W.A.P to sort the list using sort() and sorted()
- 10) W.A.P to round the value in list using round() and for loop. 11) W.A.P to convert the list into tuple. 12) W.A.P to create tuple with multiple data type. 13) W.A.P to concate the two tuple into one tuple. 14) W.A.P to access the value of index value 1st in tuple. 15) W.A.P to access the value from last in tuple. 16) W.A.P to access the value between index 1st to 5th from the tuple. 17) W.A.P to access the alternate value between index 1st to 5th.
- 18) W.A.P to create the dictionary of having 6 key and value pair. 19) W.A.P to access the value using the key from dictionary. 20) W.A.P to update the value on particular key. 21) W.A.P to separate the key and value from dictionary using keys() and values() of dictionary. 22) W.A.P to convert the two list into one dictionary using for loop. 23) W.A.P to convert the list using zip() of dictionary. 24) W.A.P to count the character repeat in string.
- Function
- Types of Function
- Function Argument
- anonymous function
- Practical Example: 1) W.A.P to print the String using the function. 2) W.A.P to create the parameterized function. 3) W.A.P to print multiple string using function. 4) W.A.P to create

calculator using function. 5) W.A.P to create lamba function using one expression. 6) W.A.P to create lamba function using two expression. 7) W.A.P to create lamba function using three expression. 8) W.A.P to create a return type function using lamda function.
- Modules
- Importing Module
- Math Module
- Random module
- Packages
- Practical Example: 1) W.A.P to import another module into one module. 2) W.A.P to use all the Math module function.

Module 15) Python - Advance python programming	10

- Printing on screen
- Reading data from keyboard
- opeaning and closing file
- reading and writing file
- Practical Example : 1) W.A.P to create the file using the python. 2) W.A.P to create a file and print the string into the file. 3) W.A.P to read a file and print the data on console. 4) W.A.P to write the multiple String into file 5) W.A.P to read multiple String from the file. 6) W.A.P to check where is the cursor in the file.
- Exception Handling
- Handling Exception
- Finally Clause
- PRactical Example: a) W.A.P to handle exception in calculator. b) W.A.P to handle multiple exception at time in one program. c) W.A.P to handle File Exception and use finally block for closing the file. d) W.A.P to print multiple exception using if else. e) W.A.P to print user define exception.
- class and object
- Attribute
- Inheritance
- Overloading
- Overriding
- Practical Example: 1) W.A.P to create a class and access the property of class using object. 2) W.A.P to create local variable and global variable. 3) W.A.P to show single inheritance. 4) W.A.P to show Multilevel inheritance. 5) W.A.P to show Multiple inheritance. 6) W.A.P to show Hierarchical inheritance. 7) W.A.P to show Hybrid inheritance. 8) W.A.P to using super() in inheritance. 9) W.A.P to show method overloading. 10) W.A.P to show Method overriding.
- sqlite3 and pymysql modules (database connectors)
- Search Function
- Match Function
- Matching Vs Searching
- Modifiers
- Practical Examples: 1) W.A.P to search a word from the string using Search() 2) W.A.P to match the word in string using Match().

- GUI Programming Introduction Tkinter programming
- Tkinter widgets
- Practical Example: 1) W.A.P to create GUI Frame. 2) W.A.P to create all the widgets using Tkinter.

Module 16) Python - DB and Python Framework	**18**

- HTML
- CSS
- javascript
- Django Introduction Advantages of django Django vs Flask
- Virtual Environment
- Project and app creation
- MVT pattern architecture
- Practical Example 1 Create Django Admin Panel 2 Creating the Doctor Finder Project. Project Practical Registration login , forgot password session management , email template , profile,updation , working with media , CRUD operations
- Practical Example: 1) Create Django Admin Panel 2) Creating the Doctor Finder Project.
- Django Admin
- URL pattern
- Template integration
- form validation using javascript
- Django Database connectivity mysql or sqllite
- ORM, Query set
- Django forms,Django authentication
- •Authentication(Sign up, login, logout, session, email sending, sms sending, otp verification, change password, forgot password, profile management)
- CRUD operationgs using AJAX
- Customization django admin panel
- Payment integration using paytm
- Github project deployment
- Live project deployment Python anywhere
- •Social authentication (For eg; Login with Google, Login with Facebook, Twitter, Github...etc) Email sending APIs (For eg; Mail chimp, Mail gun...etc.) SMS sending APIs (For eg; Twilio) Normal payments (For eg; PayPal, Stripe)
- Google Maps API

Module 17) Python - Rest Framework	**3**

- Introduction
- Requirements
- Serialization
- Requests and Responses
- Views
- URLs

- Pagination
- Settings
- Project setup
- Social Auth, Email and OTP Sending API, Third Party Integration
- RESTful API: Representational State Transfer (REST) is a widely used architectural style for building web services. Understanding REST principles and being able to create RESTful APIs is essential. CRUD API: CRUD stands for Create, Read, Update, and Delete, which are the basic operations performed on data. Creating APIs that allow these operations is fundamental to backend development. Authentication and Authorization API: Knowing how to implement user authentication and authorization mechanisms is crucia
- OpenWeatherMap API: This API provides weather data for various locations worldwide. You can retrieve current weather conditions, forecasts, and historical weather data.
- Google Maps Geocoding API: This API allows you to convert addresses into geographic coordinates (latitude and longitude) and vice versa. You can use it to retrieve location data, calculate distances between points, and display maps.
- GitHub API: GitHub provides an API that enables you to interact with repositories, issues, pull requests, and more. You can perform actions like retrieving repository information, creating issues, and accessing user data.
- Twitter API: Twitter offers an API that allows you to integrate Twitter functionality into your applications. You can fetch tweets, post tweets, retrieve user information, and perform searches.
- REST Countries API: This API provides information about countries, including details like population, languages spoken, currencies, and more. You can retrieve country-specific data and use it for various applications.
- endGrid provides an API for sending transactional and marketing emails. You can integrate it into your applications to send emails, manage templates, and track delivery statistics.
- Social authentication (For eg; Login with Google, Login with Facebook...etc)
- Email sending APIs (For eg; Mailchimp, Mailgun...etc)
- SMS sending APIs (For eg; Twilio)
- Normal payments (For eg; Paypal, Stripe)
- Google Maps API