



Microsoft Excel

Program Highlights

Microsoft Excel is the one of the most used software application of all the time. Hundreds of millions of people around the world use Microsoft Excel. You can use Excel to enter all sorts of data and perform financial mathematical or statistical calculations.

Excel Introduction

- Overview
- Various Selection Techniques
- Shortcut Keys

Customizing Excel

- Customizing the Ribbon
- Using and Customizing Autocorrect

- Changing Excel's Default Option

Using Basic Function

- Using Function- Sum, Average, Max, Min, Count, Counta
- Absolute, Mixed and Relative Referencing

Formatting and Proofing

- Formatting Cells with number formats, Alignment, Borders, etc.
- Basic Conditional

Formatting Mathematical Functions

- Sumif, Sumifs,
- Countif, Countifs,
- Averageif, Averageifs

Protecting Excel

- File Level Protection
- Workbook, worksheet protection

Text Functions

- Upper, Lower, Proper
- Left, Mid, Right
- Trim, Len, Exact
- Concatenate
- Find, Substitute

Date and Time function

- Today, Now
- Day, Month, Year
- Date, Dateif, DateAnd
- EOMonth, Weekday

Advanced Paste Special Techniques

- Paste formulas, Paste Formats
- Paste Validations
- Transpose Tables

New in Excel

- New Charts - Tree Map & waterfall
- Sunburst, box, whisker charts
- Combo Charts- Secondary Axis
- Adding Slicer Tool in Pivot & Tables

Using Power Map and Power View

- Forecast Sheet
- Sparklines - line, Column & win/Loss
- Using 3-D Map
- New Controls in pivot Table - Field, Items and sets
- Various Timelines in pivot Table
- Auto Complete a data range and list
- Quick Analysis Tool
- Smart Lookup and manage store

Sorting and Filtering

- Filtering on Text, Numbers & colors
- Sorting option
- Advanced Filter on 15-20 Different Criteria

Printing Workbooks

- Setting up Print Area
- Customizing Header and Footer
- Designing the structure of a template
- Print Titles - Repeats Rows/ Columns

What if Analysis

- Goal Seek
- Scenario Analysis
- Data Tables (PMT Function)
- Solver Tool

Logical Function

- If Function
- How to fix Errors - iferror
- Nested if
- Complex if and or function
- Charts and Slicer
- Various Charts i.e. Bar Charts / pie
- charts/Line charts
- Using Slicers , Filter data with Slicers.
- Manage Primary and Secondary Axis

Data Validation

- Number, Date & Time Validation
- Text and List Validation
- Custom Validations based on formula
- for a cell

- Dynamic Dropdown list creation using
- data Validation-Dependency list

Lookup Functions

- Vlookup / Hlookup
- Index and Match
- Creating Smooth User Interface using
- Lookup
- Nested Vlookup
- Reverse Lookup using Choose Function
- Worksheet Linking using Choose
- Function
- Worksheet Linking using Indirect
- Vlookup with Helper Column

Pivot Tables

- Creating Simple Pivot tables
- Basic and Advance Value Field Setting
- Classic Pivot table
- Grouping based on numbers and dates
- Calculated Field & Calculated items
- Arrays Functions
- What are the array Formulas, use of the
- Array Formulas
- Basic Examples of Arrays(Using ctrl+shift+enter)
- Array with if, len and mid function
- formulas
- Advanced use of Formulas with array

Excel Dashboard

- Planning a dashboard
- Adding Tables and charts to Dashboard
- Adding Dynamic Contents to
- Dashboard



POWER BI & BUSINESS ANALYTICS

About Power BI Certification Course - Overview

- ❖ Power BI is BI & Data Visualization Tool by Microsoft way advanced than SSRS by Microsoft
- ❖ According to Gartner's Magic Quadrant, it is amongst top 3 BI Tools in Industry
- ❖ Since it is a Microsoft Product it is widely used with Ms SQL Server and many other databases. Hence there is huge opportunity in Industry.
- ❖ Since it is among Top BI Tools in Industry it is one of High Paying Technologies in Industry
- ❖ Pre-requisites To Learn - none. To Implement Basic SQL is required

Address: - C-5 Basement New Krishna Park Vikaspuri Delhi – 110018
Contact No: - 8130608009, 9311646663, 01146008591

Program Curriculum

❖ Introduction to Power BI

- What is Power BI and Why Power BI
- Installing Power BI Desktop
- Exploring the Power BI Workflow
- Adjusting the settings of the Power BI Desktop
- Comparison of Power BI vs Other Reporting Tools

❖ Getting and Transforming Data with Power BI Desktop

- Connecting to different sources
- Different connecting Options (Direct Query vs Import Data Vs Live Connection)
- Shaping and transforming data with Power Query
- Editing, Merging, Appending queries etc.

❖ Modeling with Power BI

- Introduction to Modeling
- Building Relational Models (Setup and Manage Relationships)
- Creating table relationships
- Understanding the filter flow
- Cardinality and Cross filtering

❖ DAX

- Understanding Dax Syntax
- Calculated
- Columns vs Measures
- Common Dax functions and formula
- Understanding the evaluation context in DAX

❖ **Visualising Data with Reports**

- Creating Visualisations
- Color & Conditional Formatting
- Setting Sort Order
- Scatter & Bubble Charts & Play Axis
- Tooltips
- Slicers, Timeline Slicers & Sync Slicers
- Cross Filtering and Highlighting
- Visual, Page and Report Level Filters
- Drill Down/Up
- Hierarchies
- Constant Lines
- Tables, Matrices & Table Conditional Formatting
- KPI's, Cards & Gauges
- Map Visualisations
- Custom Visuals
- Managing and Arranging
- Drill through
- Custom Report Themes
- Grouping and Binning
- Bookmarks & Buttons

❖ **Introduction to Power BI Service**

- Introduction to the Power BI Service
- Quick Tour of Power BI Service
- Connecting to Data from Power BI service
- Building Blocks of Power BI Service

❖ **Sharing and Collaboration Tools in Power BI Service**

- Sharing and Collaboration Options Overview
- Publish from Power BI Desktop
- Publish Reports to Web
- Printing and Exporting from Power BI Service
- Sharing Reports & Dashboards
- Workspaces (My Workspace vs App Workspace)
- Publishing Apps in Power BI Service
- Row Level Security in Power BI Desktop and Service

❖ **Power BI Gateway and Refreshing Datasets**

- Understanding Gateways in Power BI
- Difference between Personal and On-Premise Gateway
- Installation of Personal Gateway
- Installation of On-Premise Enterprise Gateway
- Setting Up a Gateway in Power BI Service
- Understanding Data Refresh (Manual vs Scheduled Refresh) ➤
Troubleshooting Refreshing scenarios



MySQL

Program Highlights

SQL Server is a database server by Microsoft the Microsoft relational database management system is a software product which primarily stores and retrieves data requested by other applications. Therefore, a SQL Server is a database server that implements the Structured Query Language (SQL).

Program Curriculum

Introduction to SQL

- Various types of databases
- Introduction to Structured Query Language
- Distinction between client server and file server databases
- Understanding SQL Server Management Studio
- SQL Table basics
- Data types and functions
- Transaction-SQL
- Authentication for Windows
- Data control language
- The identification of the keywords in T-SQL, such as Drop Table

Database Normalization and Entity Relationship Model

- Entity-Relationship Model
- Entity and Entity Set
- Attributes and types of Attributes
- Entity Sets
- -Relationship Sets
- Degree of Relationship
- Mapping Cardinalities, One-to-One, One-to-Many, Many-to-one, Many-to-many
- Symbols used in E-R Notation

SQL Operators

- Introduction to relational databases
- Fundamental concepts of relational rows, tables, and columns
- Several operators (such as logical and relational), constraints, domains, indexes, stored procedures, primary and foreign keys
- Understanding group functions
- The unique key

Working with SQL: Join, Tables, and Variables

- Advanced concepts of SQL tables
- SQL functions
- Operators & queries
- Table creation
- Data retrieval from tables
- Combining rows from tables using inner, outer, cross, and self joins
- Deploying operators such as 'intersect,' 'except,' 'union,'

- Temporary table creation
- Set operator rules
- Table variables

Deep Dive into SQL Functions

- Understanding SQL functions – what do they do?
- Scalar functions
- Aggregate functions
- Functions that can be used on different datasets, such as numbers, characters, strings, and dates
- Inline SQL functions
- General functions
- Duplicate functions

Working with Subqueries

- Understanding SQL subqueries, their rules
- Statements and operators with which subqueries can be used
- Using the set clause to modify subqueries
- Understanding different types of subqueries, such as where, select, insert, update, delete, etc
- Methods to create and view subqueries

SQL Views, Functions, and Stored Procedures

- Learning SQL views
- Methods of creating, using, altering, renaming, dropping, and modifying views
- Understanding stored procedures and their key benefits
- Working with stored procedures
- Studying user-defined functions
- Error handling

Deep Dive into User-defined Functions

- User-defined functions
- Types of UDFs, such as scalar
- Inline table value
- Multi-statement table
- Stored procedures and when to deploy them
- What is rank function?
- Triggers, and when to execute triggers?

SQL Optimization and Performance

- Records grouping, advantages, searching, sorting, modifying data
- Clustered indexes creation
- Use of indexes to cover queries
- Common table expressions
- Index guidelines

Advanced Topics

- Correlated Subquery, Grouping Sets, Rollup, Cube, Rank, CTE, Indexes And Triggers
Correlated Subquery, Rollup, Grouping set And Cubes

Managing Database Concurrency

- Applying transactions
- Using the transaction behavior to identify DML statements
- Learning about implicit and explicit transactions
- Isolation levels management
- Understanding concurrency and locking behavior
- Using memory-optimized tables

Practice Session

- Creating Transact-SQL queries
- Querying multiple tables using joins
- Implementing functions and aggregating data
- Modifying data
- Determining the results of DDL statements on supplied tables and data
- Constructing DML statements using the output statement
- Querying data using subqueries and APPLY
- Querying data using table expressions
- Grouping and pivoting data using queries
- Querying temporal data and non-relational data
- Constructing recursive table expressions to meet business requirements
- Using windowing functions to group
- Rank the results of a query
- Creating database programmability objects by using T-SQL
- Implementing error handling and transactions
- Implementing transaction control in conjunction with error handling in stored procedures

- Implementing data types and NULL
- Designing and implementing relational database schema
- Designing and implementing indexes
- Learning to compare between indexed and included columns
- Implementing clustered index
- Designing and deploying views
- Column store views
- Explaining foreign key constraints
- Using T-SQL statements
- Usage of Data Manipulation Language (DML) Designing the components of stored procedures
- Implementing input and output parameters
- Applying error handling
- Executing control logic in stored procedures
- Designing trigger logic, DDL triggers, etc
- Accuracy of statistics
- Formulating statistics maintenance tasks
- Dynamic management objects management
- Identifying missing indexes
- Examining and troubleshooting query plans
- Consolidating the overlapping indexes
- The performance management of database instances
- SQL server performance monitoring



Python Analytics

Python Training Overview

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python has been one of the premier, flexible, and powerful open-source language that is easy to learn, easy to use, and has powerful libraries for data manipulation and analysis.

What is the Python Course Pre-requisites?

There are no hard pre-requisites. Basic understanding of Computer Programming terminologies is sufficient. Also, basic concepts related to Programming and Database is beneficial but not mandatory.

Objectives of the Course

- To understand the concepts and constructs of Python
- To create own Python programs, know the machine learning algorithms in Python and work on a real-time project running on Python.

Who should do the course?

- Big Data Professionals
- IT Developers
- Those who are showing interest to build their career in Python
- ML Engineer
- Data Scientist
- Data Analyst
- Web Developer

Python Course Curriculum

Introduction to Languages

- What is Language?
- Types of languages
- Introduction to Translators
- Compiler
- Interpreter
- What is Scripting Language?
- Types of Script
- Programming Languages v/s Scripting Languages
- Difference between Scripting and Programming

Programming Languages

- What is programming paradigm?
- Procedural programming paradigm
- Object Oriented Programming paradigm

Introduction to Python

- What is Python?
- WHY PYTHON?
- History
- Features – Dynamic, Interpreted, Object oriented, Embeddable, Extensible, Large standard libraries, Free and Open source
- Why Python is General Language?
- Limitations of Python
- What is PSF?
- Python implementations
- Python applications
- Python versions
- PYTHON IN REALTIME INDUSTRY
- Difference between Python 2.x and 3.x
- Difference between Python 3.7 and 3.8
- Software Development Architectures

Python Software's

- Python Distributions
- Download & Python Installation Process in Windows, Unix, Linux and Mac
- Online Python IDLE
- Python Real-time IDEs like Spyder, Jupyter Note Book, PyCharm, Rodeo, Visual Studio Code, ATOM, PyDev etc
- Python Language Fundamentals
- Python Implementation Alternatives/Flavors
- Keywords
- Identifiers
- Constants / Literals
- Data types
- Python VS JAVA
- Python Syntax

Different Modes of Python

- Interactive Mode
- Scripting Mode
- Programming Elements
- Structure of Python program
- First Python Application
- Comments in Python
- Python file extensions
- Setting Path in Windows
- Edit and Run python program without IDE
- Edit and Run python program using IDEs
- INSIDE PYTHON
- Programmers View of Interpreter
- Inside INTERPRETER
- What is Byte Code in PYTHON?
- Python Debugger

Python Variables

- bytes Data Type
- byte array
- String Formatting in Python
- Math, Random, Secrets Modules
- Introduction
- Initialization of variables
- Local variables

- Global variables ‘global’ keyword
- Input and Output operations
- Data conversion functions – int(), float(), complex(), str(), chr(), ord()
- Operators
- Arithmetic Operators
- Comparison Operators
- Python Assignment Operators
- Logical Operators
- Bitwise Operators
- Shift operators
- Membership Operators
- Identity Operators
- Ternary Operator
- Operator precedence
- Difference between “is” vs “==”

Input & Output Operators

- Print
- Input
- Command-line arguments

Control Statements

- Conditional control statements
- If
- If-else
- If-elif-else
- Nested-if
- Loop control statements
- for
- while
- Nested loops
- Branching statements
- Break
- Continue
- Pass
- Return
- Case studies

Data Structures or Collections

- Introduction
- Importance of Data structures
- Applications of Data structures
- Types of Collections
- Sequence
- Strings, List, Tuple, range
- Non sequence
- Set, Frozen set, Dictionary
- Strings
- What is string
- Representation of Strings
- Processing elements using indexing
- Processing elements using Iterators
- Manipulation of String using Indexing and Slicing
- String operators
- Methods of String object
- String Formatting
- String functions
- String Immutability
- Case studies

List Collection

- What is List
- Need of List collection
- Different ways of creating List
- List comprehension
- List indices
- Processing elements of List through Indexing and Slicing
- List object methods
- List is Mutable
- Mutable and Immutable elements of List
- Nested Lists
- List_of_lists
- Hardcopy, shallowCopy and DeepCopy
- zip() in Python
- How to unzip?
- Python Arrays

Tuple Collection

- What is tuple?
- Different ways of creating Tuple
- Method of Tuple object
- Tuple is Immutable
- Mutable and Immutable elements of Tuple
- Process tuple through Indexing and Slicing
- List v/s Tuple
- Case studies

Set Collection

- What is set?
- Different ways of creating set
- Difference between list and set
- Iteration Over Sets
- Accessing elements of set
- Python Set Methods
- Python Set Operations
- Union of sets
- functions and methods of set
- Python Frozen set
- Difference between set and frozenset?
- Case study

Dictionary Collection

- What is dictionary?
- Difference between list, set and dictionary
- How to create a dictionary?
- PYTHON HASHING?
- Accessing values of dictionary
- Python Dictionary Methods
- Copying dictionary
- Updating Dictionary
- Reading keys from Dictionary
- Reading values from Dictionary
- Reading items from Dictionary
- Delete Keys from the dictionary
- Sorting the Dictionary
- Python Dictionary Functions and methods

- Dictionary comprehension

Functions

- What is Function?
- Advantages of functions
- Syntax and Writing function
- Calling or Invoking function
- Classification of Functions
- No arguments and No return values
- With arguments and No return values
- With arguments and With return values
- No arguments and With return values
- Recursion
- Python argument type functions :
- Default argument functions
- Required(Positional) arguments function
- Keyword arguments function
- Variable arguments functions
- 'pass' keyword in functions
- Lambda functions/Anonymous functions
- map()
- filter()
- reduce()
- Nested functions
- Non local variables, global variables
- Closures
- Decorators
- Generators
- Iterators
- Monkey patching

Advanced Python

Python Modules

- Importance of modular programming
- What is module
- Types of Modules – Pre defined, User defined.
- User defined modules creation
- Functions based modules
- Class based modules
- Connecting modules
- Import module
- From ... import
- Module alias / Renaming module
- Built In properties of module

Packages

- Organizing python project into packages
- Types of packages – pre defined, user defined.
- Package v/s Folder
- py file
- Importing package
- PIP
- Introduction to PIP
- Installing PIP
- Installing Python packages
- Un installing Python packages OOPs
- Procedural v/s Object oriented programming
- Principles of OOP – Encapsulation , Abstraction (Data Hiding)
- Classes and Objects
- How to define class in python
- Types of variables – instance variables, class variables.
- Types of methods – instance methods, class method, static method
- Object initialization
- ‘self’ reference variable
- ‘cls’ reference variable
- Access modifiers – private(__) , protected(_), public
- AT property class
- Property() object
- Creating object properties using setaltr, getaltr

functions

- Encapsulation(Data Binding)
- What is polymorphism?
- Overriding
 1. Method overriding
 2. Constructor overriding
- Overloading
- Method Overloading
 1. Constructor Overloading
 2. Operator Overloading
- Class re-usability
- Composition
- Aggregation
- Inheritance – single , multi-level, multiple, hierarchical and hybrid inheritance and Diamond inheritance
- Constructors in inheritance
- Object class
- super()
- Runtime polymorphism
- Method overriding
- Method resolution order(MRO)
- Method overriding in Multiple inheritance and Hybrid

Inheritance

- Duck typing
- Concrete Methods in Abstract Base Classes
- Difference between Abstraction & Encapsulation
- Inner classes
- Introduction
- Writing inner class
- Accessing class level members of inner class
- Accessing object level members of inner class
- Local inner classes
- Complex inner classes
- Case studies

Exception Handling & Types of Errors

- What is Exception?
- Why exception handling?
- Syntax error v/s Runtime error
- Exception codes – AttributeError, ValueError, IndexError, TypeError...
- Handling exception – try except block
- Try with multi except
- Handling multiple exceptions with single except block
- Finally block
- Try-except-finally
- Try with finally
- Case study of finally block
- Raise keyword
- Custom exceptions / User defined exceptions
- Need to Custom exceptions
- Case studies

Regular expressions

- Understanding regular expressions
- String v/s Regular expression string
- “re” module functions
- Match()
- Search()
- Split()
- Findall()
- Compile()
- Sub()
- Subn()
- Expressions using operators and symbols
- Simple character matches
- Special characters
- Character classes
- Mobile number extraction
- Mail extraction
- Different Mail ID patterns
- Data extraction
- Password extraction
- URL extraction
- Vehicle number extraction

- Case study

File & Directory handling

- Introduction to files
- Opening file
- File modes
- Reading data from file
- Writing data into file
- Appending data into file
- Line count in File
- CSV module
- Creating CSV file
- Reading from CSV file
- Writing into CSV file
- Object serialization – pickle module
- XML parsing
- JSON parsing

Python Logging

- Logging Levels
- implement Logging
- Configure Log File in over writing Mode
- Timestamp in the Log Messages
- Python Program Exceptions to the Log File
- Requirement of Our Own Customized Logger
- Features of Customized Logger

Date & Time module

- How to use Date & Date Time class
- How to use Time Delta object
- Formatting Date and Time
- Calendar module
- Text calendar
- HTML calendar

OS module

- Shell script commands

- Various OS operations in Python
- Python file system shell methods
- Creating files and directories
- Removing files and directories
- Shutdown and Restart system

- Renaming files and directories
- Executing system commands

Python Data Base Communications (PDBC)

- Introduction to DBMS applications
- File system v/s DBMS
- Communicating with MySQL
- Python – MySQL connector
- connector module
- connect() method
- Oracle Database
- Install cx_Oracle
- Cursor Object methods
- execute() method
- executeMany() method
- fetchone()
- fetchmany()
- fetchall()
- Static queries v/s Dynamic queries
- Transaction management
- Case studies

Tkinter & Turtle

- Introduction to GUI programming
- Tkinter module
- Tk class
- Components / Widgets
- Label , Entry , Button , Combo, Radio
- Types of Layouts
- Handling events
- Widgets properties
- Case studies

NumPy for Mathematical Computing

- Introduction to mathematical computing in Python
- What are arrays and matrices, array indexing, array math, Inspecting a NumPy array, NumPy array manipulation

Pandas for Data manipulation

- What is a data Manipulation? Using Pandas library
- NumPy dependency of Pandas library
- Series object in pandas
- DataFrame in Pandas
- Loading and handling data with Pandas
- How to merge data objects
- Concatenation and various types of joins on data objects, exploring dataset

Data Visualization with Matplotlib

- Introduction to Matplotlib
- Using Matplotlib for plotting graphs and charts like Scatter, Bar, Pie, Line,
- Histogram and more
- Matplotlib API