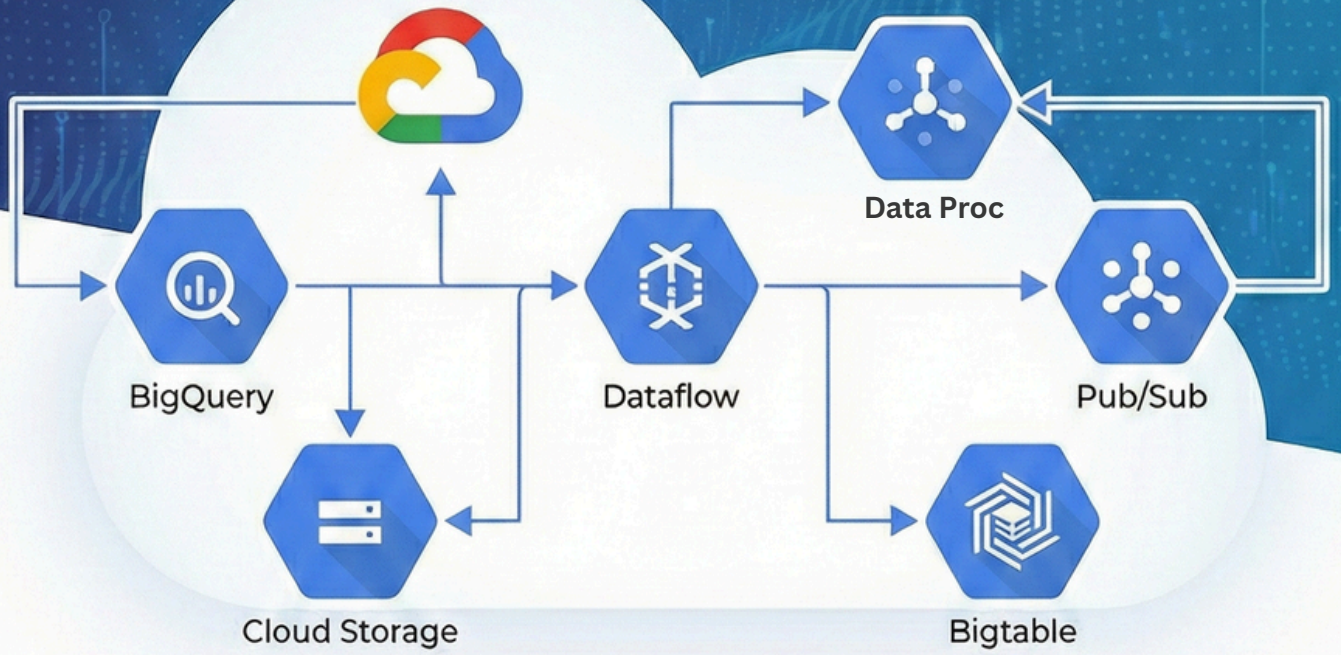
Cloud Upskill

GCP Data Engineering

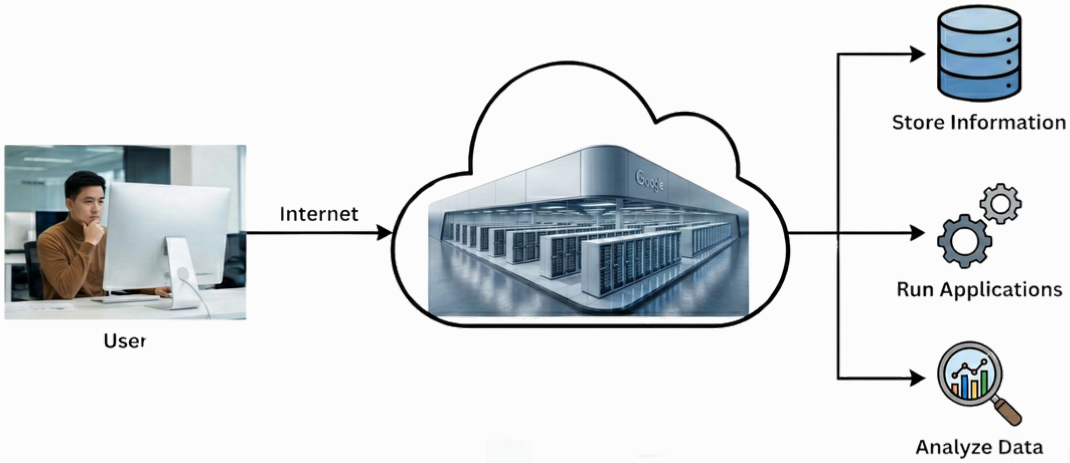
Complete Learning Path



Website
www.cloudupskills.com

What is GCP (Google Cloud Platform)?

Google Cloud Platform (GCP) is basically renting Google's powerful computers and software over the internet. It uses the exact same reliable systems that run massive services like **Google Search, YouTube, and Gmail**. Companies use GCP to store their information, run applications, and analyze data without having to build their own computer centers.



Who is a GCP Data Engineer?

A GCP Data Engineer is someone who:

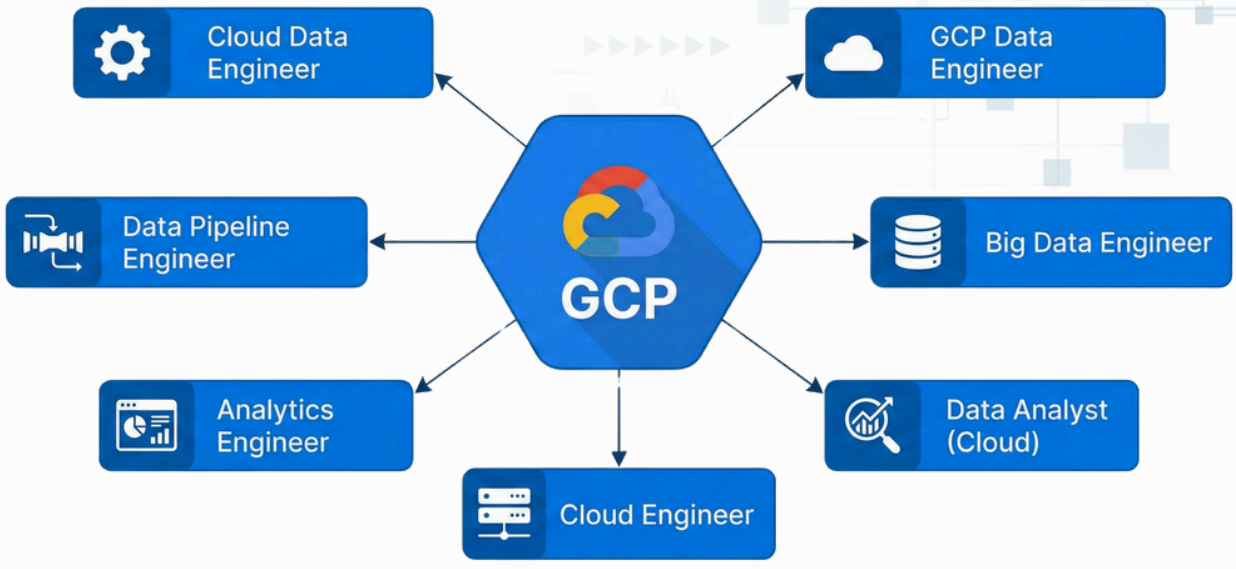
- Makes data useful for dashboards, reports & AI models
- Collects data
- Cleans data
- Stores data
- Processes data

Why Choose Cloud Upskill for GCP Data Engineering Training?

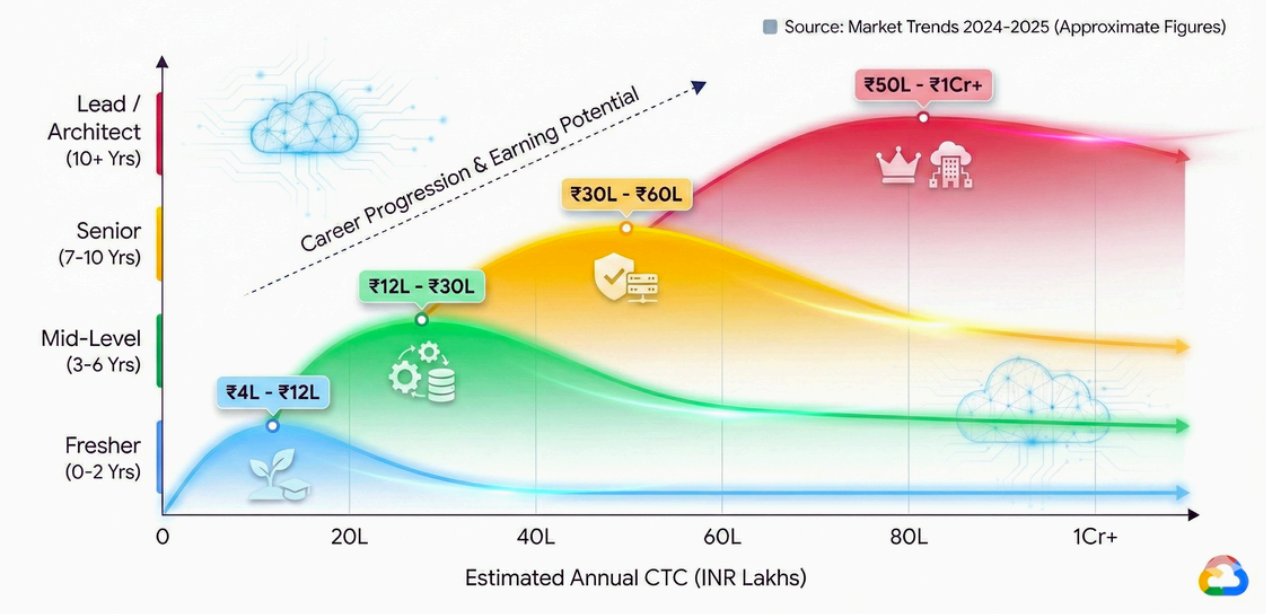
At **Cloud Upskill**, we deliver industry-oriented **GCP Data Engineering Training in Hyderabad** designed to bridge the gap between theory and real-world implementation. Our curriculum is aligned with current enterprise requirements and focuses on hands-on learning, real-time projects, and certification readiness. With expert trainers, live cloud labs, and dedicated placement support, Cloud Upskill empowers learners to build high-paying careers in cloud data engineering..

Career Opportunities after Course.

Learning **GCP Data Engineering** opens doors to multiple high-demand roles in **cloud, data, and analytics**. Companies across industries rely on data engineers to build reliable and scalable data systems.



GCP Data Engineer Salary Range in India (Fresher to Experienced)



Unlock Your Future with GCP Data Engineering

Step into the world of cloud-driven data solutions with **Cloud Upskill's GCP Data Engineering Program**. This course equips you with in-demand skills in **BigQuery, Dataflow, Dataproc, Pub/Sub, Databricks, and Cloud Composer**, helping you design scalable, secure, and high-performance data pipelines used by leading organizations worldwide.

Course Highlights – GCP Data Engineering Training at Cloud Upskill

- Industry-aligned curriculum
- Hands-on real GCP projects
- End-to-end batch and streaming pipelines
- Real-time case studies and capstone projects
- Certification-focused training (PDE & ACE)
- Resume building, Mock interviews and Placement support

Who Should Enroll?

Enroll Now – GCP Data Engineering Training at Cloud Upskill

Kick-start your journey toward a successful cloud career with **Cloud Upskill's GCP Data Engineering Course**. Gain practical expertise, industry exposure, and certification readiness with complete placement support.

Tools You'll Master in the GCP Data Engineering Program

Certification Readiness

This program prepares you for:

- **Google Cloud Professional Data Engineer (PDE)**
- **Google Cloud Associate Cloud Engineer (ACE)**

Training includes exam-oriented scenarios, architecture discussions, and hands-on practice.

Google Cloud

Google Cloud Data Engineering Training

with Real-world Projects and case Studies

Pre-Requisite: Not required, I will start the course covering all the basics keeping everyone in mind, Concepts will be cleared in both Telugu and English as needed

SQL

Introduction To DBMS

- File Management System And Its Drawbacks
- Database Management System (DBMS) and Data Models
    - Physical Data Models
    - Logical Data Models
        - Hierarchical Data Model (HDBMS)
        - Network Data Model (NDBMS)
        - Relational Data Model (RDBMS)
        - Object Data Model (ODBMS)
        - Object Relational Data Model (ORDBMS)
- Conceptual Data Models
    - Entity – Relationship (E-R) Model

Introduction To SQL Server

- Advantages and Drawbacks Of SQL Server Compared To Oracle And DB2
    - Connecting To Server
    - Server Type
    - Server Name
    - Authentication Modes
        - Sql Server Authentication Mode
        - Windows Authentication Mode

- Login and Password
- Sql Server Management Studio and Tools In Management Studio
    - Object Explorer
    - Object Explorer Details
    - Query Editor

TSQL (Transact-Structured Query Language)

Introduction To TSQL

- History and Features of TSQL
- Types Of TSQL Commands
    - Data Definition Language (DDL)
    - Data Manipulation Language (DML)
    - Data Query Language (DQL)
    - Data Control Language (DCL)
    - Transaction Control Language (TCL)
    - Database
        - Creating Database
        - Altering Database
        - Deleting Database
        - Constrains
            - Procedural Integrity Constraints
            - Declarative Integrity Constraints
                - Not Null
                - Unique
                - Default
                - Check constraints
                - Primary Key
                - Referential Integrity
                - foreign key constraints
                - Data Types In TSQL
                - Table

- Creating Table
- Altering Table
- Deleting Table

Data Manipulation Language

- Insert
  - Identity
  - Creating A Table From Another Table
  - Inserting Rows From One Table To Another
  - Update
    - Computed Columns
    - Delete
    - Truncate
    - Differences Between Delete and Truncate

Data Query Language (DQL)

- Select
- Where clause
- Order By Clause
- Distinct Keyword
- Isnull() function
- Column aliases
- Predicates
  - Between … And
  - In
  - Like
  - Is Null

Built In Functions

- Scalar Functions
  - Numeric Functions
  - Character Functions
  - Conversion Functions
  - Date Functions

- Aggregate Functions
    - COUNT
    - SUM
    - AVG
    - MIN,
    - MAX

- Convenient Aggregate Functions
- Statistical Aggregate Functions
- Group By and Having Clauses
- Super Aggregates
- Over(partition by ...) Clause
- Ranking Functions
- Common Table Expressions (CTE)

Set Operators

- Union
- Intersect
- Except

Joins

- Inner Join
    - Equi Join
    - Natural Join
    - Non-Equi Join
    - Self Join
    - Outer Join
        - Left Outer Join
        - Right Outer Join
        - Full Outer Join
        - Cross Join

Sub Queries

- Single Row Sub Queries
- Multi Row Sub Queries
    - Any or Some
    - ALL
    - Nested Sub Queries
    - Co-Related Sub Queries
        - Exists and Not Exists

Indexes

- Clustered Index
- NonClustered Index
- Create
- Alter
- Drop Indexes
- Using Indexes

Security

- Login Creation
    - SQL Server Authenticated Login
    - Windows Authenticated Login
    - User Creation
    - Granting Permissions
    - Revoking Permissions
    - Roles

Views

- Purpose Of Views
- Creating
- Altering
- Dropping Indexes
- Simple and Complex Views
- Encryption
- Schema Binding Options in creating views

Transaction Management

- Introduction
- Begin Transaction
- Commit Transaction
- Rollback Transaction
- Save Transaction
- Role Of Log File In Transaction Management
- Implicit Transactions

TSQL Programming

- Drawbacks Of TSQL that leads to TSQL Programming
- Introduction To TSQL Programming
- Control statements In TSQL Programming
    - Conditional Control Statements
        - If
        - Case
    - Looping Control Statements
        - While

Cursors

- Working With Cursors
- Types Of Cursors
    - Forward_Only and Scroll Cursors
    - Static Cursors
    - Dynamic Cursors
    - Keyset Cursors
    - Local Cursors
    - Global Cursors

Stored Sub Programs

- Advantages Of Stored Sub Programs compared to Independent SQL Statements

- Stored Procedures
    - Creating
    - Altering
    - Dropping
    - Optional Parameters
    - Input Parameters
    - Output Parameters
    - Permissions on Stored Procedures

User Defined Functions

- Creating, Altering and Dropping
- Types Of User Defined Functions
    - Scalar Functions
    - Table Valued Functions
        - Inline Table Valued Functions
        - Multi Statement Table Valued Functions
- Permissions On User Defined Functions

Triggers

- Purpose of Triggers
- Differences Between Stored Procedures and User Defined Functions and Triggers
- Creating
- Altering
- Dropping Triggers
- Magic Tables
- Instead Of Triggers

Exception Handling

- Implementing Exception Handling
- Adding and removing User Defined Error Messages To And From SQL Server Error Messages List
- Raising Exceptions Manual

Working With XML Data Type
Attach and Detach of Database

Python

Introduction to Python

- What is Python?
- WHY PYTHON?
- History
- Features
    - Dynamic
    - Interpreted
- Object oriented
- Embeddable
- Extensible
- Large standard libraries
- Free Source
- Open source
- Why Python is General Language?
- Limitations of Python
- What is PSF?
- Python implementations
- Python applications
- Python versions
- Python in Real Time Industry
- Software Development Architectures

Python Software's

- Python Distributions
- Download & Python Installation Process
    - Windows
    - Unix
    - Linux
    - Mac
- Online Python IDLE
- Python Real-time IDEs
    - Spyder
    - Jupyter Note Book
    - PyCharm
    - Rodeo,
    - Visual Studio Code
    - ATOM
    - PyDevetc

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 operations
- 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

Data Structures or Collections

- 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
- DeepCopy
- zip() in Python
- How to unzip?
- Python Arrays
- Case studies

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

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
    - Method overriding
    - Constructor overriding
- Overloading
    - Method Overloading
    - Constructor Overloading
- Operator Overloading
    - Class re-usability
    - Composition
    - Aggregation
    - Inheritance
        - single
        - multi-level
        - multiple
        - hierarchical
        - hybrid inheritance
        - Diamond inheritance
- Constructors in inheritance
- Object class
- super()
- Runtime polymorphism
- Method overriding
- Method resolution order(MRO)
- Method overriding in Multiple inheritance
- 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
- What is Exception?
- Why exception handling?
- Syntax error v/s Runtime error
- Exception codes
- 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

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

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

Multi-threading & Multi Processing

- Introduction
- Multi tasking v/s Multi threading
- Threading module
- Creating thread – inheriting Thread class
- Using callable object
- Life cycle of thread
- Single threaded application
- Multi threaded application

- Can we call run() directly?
- Need to start() method
- Sleep()
- Join()
- Synchronization
    - Lock class
    - acquire()
    - release() functions

Garbage collection

- Introduction
- Importance of Manual garbage collection
- Self reference objects garbage collection
- 'gc' module
- Collect() method
- Threshold function

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 method
- execute() method
- executeMany() method
- fetchone()
- fetchmany()
- fetchall()
- Static queries v/s Dynamic queries
- Transaction management

Python – Network Programming

- What is Sockets?
- What is Socket Programming?
- The socket Module
- Server Socket Methods
- Connecting to a server
- A simple server-client program
- Server
- Client

Tkinter & Turtle

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

Data analytics modules

- Numpy
- Introduction
- Scipy
- Introduction
- Arrays
- Datatypes
- Matrices
- N dimension arrays
- Indexing and Slicing
- Pandas
- Data Frames

- Merge
- Join
- Concat
- MatPlotLib introduction
- Drawing plots

DJANGO

- Introduction to PYTHON Django
- What is Web framework?
- Why Frameworks?
- Define MVT Design Pattern
- Difference between MVC and MVT

PANDAS

- Introduction to Pandas
- Environment Setup Pandas
- Introduction to Data Structures
    - Dimension & Description
    - Series
    - DataFrame
    - Data Type of Columns
    - Panel

Pandas — Series

- Series
- Create an Empty Series
- Create a Series f
- rom ndarray
- rom dict
- rom Scalar
- Accessing Data from Series with Position
- Retrieve Data Using Label (Index)

Pandas – DataFrame

- DataFrame
- Create DataFrame
- Create an Empty DataFrame
- Create a DataFrame from Lists
- Create a DataFrame from Dict of ndarrays / Lists
- Create a DataFrame from List of Dicts
- Create a DataFrame from Dict of Series
- Column Selection
- Column Addition
- Column Deletion
- Row Selection
- Addition
- Deletion

Pandas – Panel

- Panel()
- Create Panel
- Selecting the Data from Panel

Pandas – Basic Functionality

- DataFrame Basic Functionality

Pandas – Descriptive Statistics

- Functions & Description
- Summarizing Data

Pandas – Function Application

- Table-wise Function Application
- Row or Column Wise Function Application
- Element Wise Function Application

Pandas – Reindexing

- Reindex to Align with Other Objects
- Filling while ReIndexing
- Limits on Filling while Reindexing
- Renaming

Pandas – Iteration

- Iterating a DataFrame
- iteritems()
- iterrows()
- itertuples()

Pandas – Sorting

- By Label
- Sorting Algorithm

Pandas – Working with Text Data

Pandas – Options and Customization

- get_option(param)
- set_option(param,value)
- reset_option(param)
- describe_option(param)
- option_context()

Pandas – Indexing and Selecting Data

- .loc()
- .iloc()
- .ix()
- Use of Notations

Pandas – Statistical Functions

- Percent_change
- Covariance
- Correlation
- Data Ranking

Pandas – Window Functions

- .rolling() Function
- .expanding() Function
- .ewm() Function

Pandas – Aggregations

- Applying Aggregations on DataFrame

Pandas – Missing Data

- Cleaning / Filling Missing Data
- Replace NaN with a Scalar Value
- Fill NA Forward and Backward
- Drop Missing Values
- Replace Missing (or) Generic Values

Pandas – GroupBy

- Split Data into Groups
- View Groups
- Iterating through Groups
- Select a Group
- Aggregations
- Transformations
- Filtration

Pandas – Merging/Joining

- Merge Using 'how' Argument

Pandas – Concatenation

- Concatenating Objects
- Time Series

Pandas – Date Functionality
Pandas – Timedelta
Pandas – Categorical Data

- Object Creation

Pandas – Visualization

- Bar Plot
- Histograms
- Box Plots

- Area Plot
- Scatter Plot
- Pie Chart

Pandas – IO Tools

- Pandas – IO Tools

Pandas – Sparse Data
Pandas – Caveats & Gotchas
Pandas – Comparison with SQL

NUMPY

- NUMPY - INTRODUCTION
- NUMPY - ENVIRONMENT
- NUMPY - NDARRAY OBJECT
- NUMPY - DATA TYPES
    - Data Type Objects (dtype)
- NUMPY - ARRAY ATTRIBUTES
    - shape
    - ndim
    - itemsize
    - flags
- NUMPY - ARRAY CREATION ROUTINES
    - empty
    - zeros
    - ones
- NUMPY - ARRAY FROM EXISTING DATA
    - asarray
    - frombuffer
    - fromiter

- NUMPY - ARRAY FROM NUMERICAL RANGES
    - arange
    - linspace
    - logspace
- NUMPY - INDEXING & SLICING
- NUMPY - ADVANCED INDEXING
    - Integer Indexing
    - Boolean Array Indexing
- NUMPY - BROADCASTING
- NUMPY - ITERATING OVER ARRAY
    - Iteration
    - Order
    - Modifying Array Values
    - External Loop
    - Broadcasting Iteration
- NUMPY – ARRAY MANIPULATION
    - reshape
    - ndarray.flat
    - ndarray.flatten
    - ravel
    - transpose
    - ndarray.T
    - swapaxes
    - rollaxis
    - broadcast
    - broadcast_to
    - expand_dims
    - squeeze
    - concatenate
    - stack
    - hstack and numpy.vstack
    - split

- hsplit and numpy.vsplit
- resize
- append
- insert
- delete
- unique

- NUMPY – BINARY OPERATORS
  - bitwise_and
  - bitwise_or
  - invert()
  - left_shift
  - right_shift

- NUMPY - STRING FUNCTIONS
- NUMPY - MATHEMATICAL FUNCTIONS
  - Trigonometric Functions
  - Functions for Rounding

- NUMPY - ARITHMETIC OPERATIONS
  - reciprocal()
  - power()
  - mod()

- NUMPY - STATISTICAL FUNCTIONS
  - amin() and numpy.amax()
  - ptp()
  - percentile()
  - median()
  - mean()
  - average()
  - Standard Deviation
  - Variance

- NUMPY - SORT, SEARCH & COUNTING FUNCTIONS
    - sort()
    - argsort()
    - lexsort()
    - argmax() and numpy.argmin()
    - nonzero()
    - where()
    - extract()
- NUMPY - BYTE SWAPPING
    - ndarray.byteswap()
- NUMPY - COPIES & VIEWS
    - No Copy
    - View or Shallow Copy
    - Deep Copy
- NUMPY - MATRIX LIBRARY
    - empty()
    - matlib.zeros()
    - matlib.ones()
    - matlib.eye()
    - matlib.identity()
    - matlib.rand()
- NUMPY - LINEAR ALGEBRA
    - dot()
    - vdot()
    - inner()
    - matmul()
    - Determinant
    - linalg.solve()

- NUMPY - MATPLOTLIB
  - Sine Wave Plot
  - subplot()
  - bar()
- NUMPY – HISTOGRAM USING MATPLOTLIB
  - histogram()
  - plt()
- NUMPY - I/O WITH NUMPY
  - save()
  - savetxt()

GCP Introduction

- Understand what Google Cloud Platform (GCP) does in simple terms.
- Overview of Google Cloud Platform services.
- Learn how cloud computing helps save time and cost.
- Create a free Google Cloud account step by step
- Data engineer roles and responsibilities
- GCP resource hierarchy: organizations, folders, projects
- Regions, zones and multi-region concepts

GCP Interfaces and Management Tools

Google Cloud Console

1. Introduction to Cloud Console
   - What is Google Cloud Console
   - Role of Cloud Console in GCP
   - When to use Console vs CLI

2. Navigating the Cloud Console

- Projects & Project Selector
- Dashboard Overview
- Search Bar & Service Navigation

3. Resource Management

- Creating & Managing Projects
- Enabling GCP Services
- Managing Quotas & Limits

4. Identity & Access Management (IAM)

- Users, Roles & Permissions
- Service Accounts
- Best Practices for Access Control

5. Data Engineer Hands-On Tasks

- Creating GCS Buckets
- Creating BigQuery Datasets & Tables
- Running Queries in BigQuery UI
- Monitoring Dataflow & Dataproc Jobs

6. Monitoring & Logging

- Cloud Monitoring Dashboards
- Viewing Logs
- Alerting Basics

Cloud Shell

1. Introduction to Cloud Shell

- What is Cloud Shell
- Advantages of Cloud Shell
- Cloud Shell Environment Overview

2. Cloud Shell Tools
    - Pre-installed Tools
        - gcloud
        - gsutil
        - bq
        - Python & Git
3. Basic Cloud Shell Commands
    - Navigating Files & Directories
    - Running GCP Commands
    - Uploading & Downloading Files
4. Data Engineer Hands-On Tasks
    - Creating GCS Buckets using gsutil
    - Uploading Data to GCS
    - Running BigQuery Queries using bq
    - Executing Python ETL Scripts
5. Automation & Scripting
    - Writing Shell Scripts
    - Scheduling Jobs
    - Using Environment Variables

Google Cloud SDK

1. What is Google Cloud SDK?
    - Official toolkit to interact with GCP
    - Used to manage services from command line and code
    - Installed on local machines or available via Cloud Shell
2. Components of Google Cloud SDK
    - gcloud
    - gsutil
    - bq

3. Installing Google Cloud SDK

    - SDK installation on Windows
    - SDK installation on Linux & macOS
    - Initial setup & authentication
    - Setting default project & region

4. Using SDK for Data Engineering Tasks

    - Creating projects & enabling services
    - Uploading data to GCS
    - Running BigQuery queries
    - Submitting Dataflow & Dataproc jobs

Command Line Interface (CLI Tools)

1. gcloud CLI

    - Authentication & Configuration
    - Managing Projects & Services
    - IAM Management via CLI

2. gsutil

    - File Upload & Download
    - Bucket Management
    - Lifecycle Rules

3. bq CLI

    - Running SQL Queries
    - Loading & Extracting Data
    - Managing Datasets & Tables

GCP Interfaces and Management Tools

1. **Introduction to GCP Locations**
    - What are cloud locations?
    - Why location selection matters in GCP
    - Real-world business impact of location choice

Linux Basics on Cloud Shell

- Getting started with Linux
- Linux Installation
- Basic Linux Commands

- Cloud shell tips
- File and Directory Operations
    - ls
    - cd
    - pwd
    - mkdir
    - rmdir
    - cp
    - mv
    - touch
    - rmnano

- File Content Manipulation
    - cat
    - less
    - head
    - tail
    - grep

- Text Processing
    - awk
    - sed
    - cut
    - sort
    - uniq

- User and Permission related
    - whoami
    - id
    - su
    - sudo
    - chmod
    - chown

Google Cloud Storage

- Overview of Cloud Storage.
- Understanding buckets
- Objects in Cloud Storage.
- Use cases for Cloud Storage
- Creating and managing Cloud Storage buckets.
- Uploading and downloading objects to and from Cloud Storage.
- Setting access controls and permissions for buckets and objects.
- Data Transfer and Lifecycle Management.
- Versioning and Object Versioning.
- Integration with Other GCP Services.
- Implementing best practices for optimizing Cloud Storage performance.
- Securing data in Cloud Storage with encryption and access controls.
- Monitoring and logging for Cloud Storage operations.

Cloud SQL

- Introduction to Cloud SQL
- Creating and Managing Cloud SQL Instances
- Configuring database settings, users, and access controls.
- Connecting to Cloud SQL instances using Cloud SQL studio, Shell, Workbenches
- Importing and exporting data in Cloud SQL.
- Backups and High Availability
- Integration with Other GCP Services
- Managing database user roles and permissions.
- Introduction to DMS
- End to End Database migration Project
    - Offline: Export and Import method
    - Online: DMS method

BigQuery (SQL Development)

- Query (SQL Development)
- Introduction to BigQuery
- BigQuery Architecture
- Use cases for BigQuery in business intelligence and analytics.

- Various method of creating table in BigQuery
- BigQuery Data Sources and File Formats
- Native table and External Tables
- SQL Queries and Performance Optimization
    - Writing and optimizing SQL queries in BigQuery.
    - Understanding query execution plans and best practices.
    - Partitioning and clustering tables for performance.
- Data Integration and Export
    - Loading data into BigQuery from Cloud Storage, Cloud SQL, and other sources.
    - Exporting data from BigQuery to various formats.
    - Real-time data streaming into BigQuery

- Configuring access controls and permissions in BigQuery.
- BigQuery Views:
    - Views
    - Materialized Views
    - Authorized Views
- Integration with Other GCP Services
    - Integrating BigQuery with Dataflow for ETL processes.
    - Building data pipelines with BigQuery and Composer.
- Case Study-1: Spotify
- Case Study-2: Social Media

DataProc (Pyspark Development)

- Introduction to Hadoop and Apache Spark
- Understanding the difference between Spark and MapReduce
- What is Spark and Pyspark.
- Understanding Spark framework and its functionalities.
- Overview of DataProc as a fully managed Apache Spark and Hadoop service.
- Use cases for DataProc in data processing and analytics.
- Cluster Creation and Configuration
    - Creating and managing DataProc clusters.
    - Configuring cluster properties for performance and scalability.
    - Preemptible instances and cost optimization

- Running Jobs on DataProc
    - Submitting and monitoring Spark and Hadoop jobs on DataProc.
    - Use of initialization actions and custom scripts.
    - Job debugging and troubleshooting.
- Integration with Storage and BigQuery
    - Reading and writing data from/to Cloud Storage and BigQuery.
    - Integrating DataProc with other storage solutions.
    - Performance optimization for data access
- Automation and scheduling of recurring jobs.
- Case Study-1: Data Cleaning of Employee Travel Records.
- End to End Batch Pyspark pipeline using Dataproc, BigQuery, GCS

Databricks on GCP

- What is Databricks lakehouse platform.
- Databricks architecture and components.
- Setting up and Administering a Databricks workspace
- Managing data with Delta Lake Databricks Unity Catalog
- Note books and clusters
- ELT with Spark SQL and Python
- optimize performance within Databricks.
- Incremental Data Processing
- Delta Live tables
- Case study: creating end to end workflows

DataFlow (Apache Beam development)

- Introduction to DataFlow
- Use cases for DataFlow in real-time analytics and ETL.
- Understanding the difference between Apache Spark and Apache Beam
- How Dataflow is different from Dataproc
- Building Data Pipelines with Apache Beam
    - Writing Apache Beam pipelines for batch and stream processing.
    - Custom Pipelines and Pre-defined pipelines.
    - Transformations and windowing concepts.
- Integration with Other GCP Services
    - Integrating DataFlow with BigQuery, Pub/Sub, and other GCP services.
    - Real-time analytics and visualization using DataFlow and BigQuery.
    - Workflow orchestration with Composer.
- End to End Streaming Pipeline using Apache beam with Dataflow, Python app, PubSub, BigQuery, GCS.
- Template method of creating pipelines

Cloud Pub/Sub

- Introduction to Pub/Sub
- Understanding the role of Pub/Sub in event-driven architectures.
- Key Pub/Sub concepts: topics, subscriptions, messages, and acknowledgments.
- Creating and Managing Topics and Subscriptions
    - Using the GCP Console to create Pub/Sub topics and subscriptions.
    - Configuring message retention policies and acknowledgment settings.

- Publishing and Consuming Messages
    - Writing and deploying code to publish messages to a topic.
    - Implementing subscribers to consume and process messages from subscriptions.
- Integration with Other GCP Services
    - Connecting Pub/Sub with Cloud Functions for serverless event-driven computing.
    - Integrating Pub/Sub with Dataflow for real-time stream processing.
- Streaming use-case using Dataflow

Cloud Composer (DAG Creations)

- Introduction to Composer/Airflow
- Overview of Airflow Architecture
- Use cases for Composer in managing and scheduling workflows.
- Creating and Managing Workflows
    - Creating and configuring Composer environments.
    - Defining and scheduling workflows using Apache Airflow.
    - Monitoring and managing workflow executions.
- Integration with Data Engineering Services
    - Orchestrating workflows involving BigQuery, DataFlow, and other services.
    - Coordinating ETL processes with Composer.
    - Integrating with external systems and APIs
- Error Handling and Troubleshooting
    - Handling errors and retries in Composer workflows.
    - Debugging and troubleshooting failed workflow executions.
    - Logging and monitoring for Composer workflows

- Level-1-DAG: Orchestrating the BigQuery pipelines
- Level-2-DAG: Orchestrating the DataProc pipelines
- Level-3-DAG: Orchestrating the Dataflow pipelines
- Implementing CI/CD in Composer Using Cloud Build and GitHub

Data Fusion

- Introduction to Data Fusion
    - Overview of Data Fusion as a fully managed data integration service.
    - Use cases for Data Fusion in ETL and data migration.
- Building Data Integration Pipelines
    - Creating ETL pipelines using the visual interface.
    - Configuring data sources, transformations, and sinks.
    - Using pre-built templates for common integration scenarios
- Integration with GCP and External Services
    - Integrating Data Fusion with BigQuery, Cloud Storage, and other GCP services
- End to End pipeline using Data fusion with Wrangler, GCS, BigQuery

Cloud Functions

- Cloud Functions Introduction
- Setting up Cloud Functions in GCP
- Event-driven architecture and use cases
- Writing and deploying Cloud Functions
- Triggering Cloud Functions:
    - HTTP triggers
    - Pub/Sub triggers
    - Cloud Storage triggers

- Monitoring and logging Cloud Functions
- Usecase-1: Loading the files from GCS to BigQuery as soon as it is uploaded

Terraform

- Terraform Introduction
- Installing and configuring Terraform.
- Infrastructure Provisioning
- Terraform basic commands
    - Init
    - plan
    - apply
    - destroy
- Create Resources in Google Cloud Platform
    - GCS buckets
    - Dataproc cluster
    - BigQuery Datasets and tables
    - And more resources as needed

By the End of the course What Students can Expect

Proficient in SQL Development:

- Mastering SQL for querying and manipulating data within Google BigQuery and Cloud SQL.
- Writing complex queries and optimizing performance for large-scale datasets.
- Understanding schema design and best practices for efficient data storage.

Pyspark Development Skills:

- Proficiency in using PySpark for large-scale data processing on Google Cloud.
- Developing and optimizing Spark jobs for distributed data processing.

- Understanding Spark's RDDs, Dataframes, and transformations for data manipulation.

Pyspark Development Skills:

- Proficiency in using PySpark for large-scale data processing on Google Cloud.
- Developing and optimizing Spark jobs for distributed data processing.
- Understanding Spark's RDDs, Dataframes, and transformations for data manipulation.

Apache Beam Development Mastery:

- Creating data processing pipelines using Apache Beam.
- Understanding the concepts of parallel processing and data parallelism.
- Implementing transformations and integrating with other GCP services.

DAG Creations with Cloud Composer:

- Designing and implementing Directed Acyclic Graphs (DAGs) for orchestrating workflows.
- Using Cloud Composer for workflow automation and managing dependencies.
- Developing DAGs that integrate various GCP services for end-to-end data processing.

Notebooks, Workflows with Databricks:

- Understand how to build and manage data pipelines using Databricks and Delta Lake.
- Efficiently query and analyze large datasets with Databricks SQL and Apache Spark.
- Implement scalable workflows and optimize performance within Databricks

Architecture Planning:

- Proficient in architecting end-to-end data solutions on GCP.
- Understanding the principles of designing scalable, reliable, and cost-effective data architectures.

This course is designed to equip students with strong, hands-on expertise in SQL, PySpark, Apache Beam, DAG creation, and data architecture planning. By focusing on real-world use cases and industry best practices, the program prepares learners to confidently address modern data engineering challenges and successfully pursue Google Cloud Platform (GCP) certifications.

Topics Covered:
The curriculum includes comprehensive coverage of key services and tools related to the modules mentioned above, with practical implementation throughout the course.

Course Duration:
60+ hours in total, with daily sessions lasting 1 to 1.5 hours.

Google Cloud

About Cloud Upskill

Cloud Upskill is one of the leading training institutes in Hyderabad, offering certified, job-oriented training with live projects, workshops, and hands-on practice. With over 10+ years of experience, the institute provides online training delivered by certified, industry-expert trainers.

High Industry Demand

Our courses are designed with current market requirements, so our students stay in demand across top companies.

100% Student Satisfaction

Our learners consistently rate our training 5-star for teaching quality, support, and real-world applicability.

Expert Trainers & Practical Learning

Every program is led by experienced professionals and includes hands-on projects, assignments, and mentoring to build real skills.

Proven Track Record

Delivering industry-focused training and successful careers for nearly 10 years with strong placement outcomes.

Google Cloud

Key Highlights

1. Free Tablet
2. Lifetime Recordings
3. Course Materials
4. Weekly Assessment Tests
5. Mock Interviews
6. Resume Building Support
7. 100% Placement Assistance
8. Hands-on Real Projects
9. Course Completion Certificate

Cloud Upskill

Contact Us :

Reach out to our dedicated team for any inquiries, assistance, or information you need.

info@cloudupskills@gmail.com

Prestige Sky Tech, Financial District, Nanakramguda, Hyderabad, Telangana 500032

www.cloudupskills.com

+91 - 8019953358