CareerByteCode
Learning Made simple

**CareerByteCode Admissions**
support@careerbytecode.in
www.careerbytecode.in



CAREERBYTECODE PROGRAM

# DataFoundry:
# Cloud & Big Data Engineering

The CareerByteCode DataFoundry - Cloud & Big Data Engineering is an eight-week practical program for learners with basic Python and SQL knowledge who want to build skills in data ingestion, transformation, storage, processing and orchestration.

## Build data pipelines.
## Run them in production.

Mentors: Jenefer Rexee George & Daisy Grace Thomas

8 weeks · 24 live classes · 48 instructor-led hours · 4 major projects · Hybrid: classroom + online

# Course Overview

LEARNING MADE SIMPLE

The CareerByteCode DataFoundry - Cloud & Big Data Engineering is an eight-week practical program for learners with basic Python and SQL knowledge who want to build skills in data ingestion, transformation, storage, processing and orchestration. Learners progress from Python-based data handling and advanced SQL to ETL/ELT design, data lake and warehouse concepts, Apache Spark/PySpark, Apache Airflow, Kafka fundamentals and cloud data services.

The program focuses on building reliable batch data pipelines and understanding the foundations of modern cloud data platforms. Each batch will use one selected cloud provider - AWS, Azure or Google Cloud - for the hands-on cloud component. The provider and equivalent services will be confirmed before the batch begins.

| 8 | 24 | 48 | 16 | 4 |
|---|---|---|---|---|
| WEEKS | LIVE CLASSES | HOURS | MODULES | PROJECTS |

# Your Mentors

LEARN FROM PRACTITIONERS



### Jenefer Rexee George

AI ML DATA CONSULTANT

Jenefer Rexee George is an AI and ML Data Consultant and Software Engineer (AI/ML) who builds intelligent systems that combine data, machine learning and modern AI to solve real-world problems.



### Daisy Grace Thomas

AI CONSULTANT · AI PRODUCT & AUTOMATION ENGINEER

Daisy Grace Thomas is an AI Product and Automation Engineer and Business Analyst with 9+ years delivering enterprise solutions across IT services, EdTech, finance and aerospace.

CareerByteCode
Learning Made simple

# Table of Contents

DATAFOUNDRY DATA ENGINEERING


01 Program at a Glance

02 Who This Course Is for

03 Before You Start: Prerequisites

04 What You'll Achieve: Learning Outcomes

05 Week-Wise Curriculum Overview

06 Four Major Hands-On Projects

07 Tools and Technologies

08 Assessment and Evaluation

09 Training Methodology and Delivery

10 Career Support and Portfolio

11 Commercial Details

12 Important Academic and Operational Notes


**How every class works**

Each class combines concept explanation, a live instructor demonstration, guided practice and a short recap. You practise between classes and finish four projects for your portfolio. Delivery: Hybrid, classroom + online. Next batch dates are shared on enrolment.


CareerByteCode

Learning Made simple

# Program at a Glance

DATAFOUNDRY DATA ENGINEERING

| | |
|---|---|
| **Duration** | 8 weeks |
| **Schedule** | 3 classes per week; 24 classes total |
| **Class duration** | 2 hours |
| **Instructor-led learning** | 48 hours |
| **Target audience** | Students, freshers, working IT professionals, software developers and Data Analysts |
| **Entry level** | Basic Python and SQL knowledge; limited Data Engineering experience |
| **Core programming** | Python and SQL |
| **Processing** | Apache Spark and PySpark |
| **Orchestration** | Apache Airflow |
| **Streaming** | Apache Kafka fundamentals |
| **Data architecture** | Data Lake, Data Warehouse and Lakehouse fundamentals |
| **Cloud** | One provider per batch: AWS, Azure or Google Cloud |
| **Projects** | 4 major hands-on projects |
| **Delivery** | Hybrid: classroom + online |
| **Course fee** | ₹40,000 + applicable GST |

CareerByteCode
Learning Made simple

# Who This Course Is for

DATAFOUNDRY DATA ENGINEERING

Students, freshers, working IT professionals, software developers and Data Analysts

# Before You Start: Prerequisites

DATAFOUNDRY DATA ENGINEERING

- Basic Python knowledge: variables, data types, conditions, loops, functions and basic collections.
- Basic SQL knowledge: SELECT, WHERE, GROUP BY, aggregate functions and simple joins.
- Basic familiarity with relational databases and tables.
- Comfort using a computer, terminal or command prompt and a web browser.
- A laptop with internet access and permission to install or run course tools.
- Prior cloud experience is not mandatory; cloud data engineering fundamentals will be introduced during the course.
- Learners should be prepared to practise between classes and complete guided assignments.

# What You'll Achieve: Learning Outcomes

DATAFOUNDRY DATA ENGINEERING

By the end of the program, learners should be able to:

- Explain the role of a Data Engineer and the end-to-end data lifecycle.

- Use Python to read, validate, transform and write CSV, JSON and API-sourced data.
- Write intermediate and advanced SQL queries using joins, CTEs, window functions and aggregations.
- Explain ETL vs ELT, batch vs streaming and common data pipeline design patterns.
- Design basic relational and analytical data models, including star-schema concepts.
- Explain Data Lake, Data Warehouse and Lakehouse architectures and their typical use cases.
- Use Apache Spark and PySpark DataFrames to process and transform larger datasets.
- Build and schedule basic Apache Airflow DAGs with dependencies, retries and logging.
- Explain Kafka topics, producers, consumers, partitions and consumer groups at a foundational level.
- Use one selected cloud provider's storage, identity and data-processing services in a guided pipeline.
- Apply basic data quality, validation, monitoring and failure-handling practices.
- Present an end-to-end data engineering project with architecture, code, pipeline flow and documentation.

CareerByteCode
Learning Made simple

# Week-Wise Curriculum Overview

DATAFOUNDRY DATA ENGINEERING

| WEEK | FOCUS | MODULES | CLASSES |
|---|---|---|---|
| 1 | Data Engineering Foundations and Python for Data | 2 | Classes 1-3 |
| 2 | Advanced SQL and Data Modelling | 2 | Classes 4-6 |
| 3 | Data Ingestion, ETL/ELT and Pipeline Development | 2 | Classes 7-9 |
| 4 | Data Lakes, Warehouses and Lakehouse Architecture | 2 | Classes 10-12 |
| 5 | Apache Spark and PySpark | 2 | Classes 13-15 |
| 6 | Apache Airflow and Workflow Orchestration | 2 | Classes 16-18 |
| 7 | Kafka Fundamentals and Cloud Data Engineering | 2 | Classes 19-21 |
| 8 | Data Quality, Monitoring and End-to-End Capstone | 2 | Classes 22-24 |

The course contains 16 modules (2 per week) and 24 classes (3 per week). Each class is planned for two hours, with a balance of explanation, instructor demonstration, guided coding and recap.

## WEEK 01 — Data Engineering Foundations and Python for Data

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

### Module 1: Introduction to Data Engineering

- What is Data Engineering and what does a Data Engineer do?
- End-to-end data lifecycle: source, ingest, store, process, serve and consume
- Batch processing vs real-time/stream processing
- Overview of operational databases, data lakes and data warehouses
- Data Engineering vs Data Analytics vs Data Science
- Structured, semi-structured and unstructured data
- ETL vs ELT and common pipeline patterns

CareerByteCode
Learning Made simple

### Module 2: Python for Practical Data Processing

- Python refresh: functions, collections, modules and exceptions
- Reading and writing CSV, JSON and text files
- Calling REST APIs and handling JSON responses
- Virtual environments, packages and configuration/secrets hygiene
- Working with files and directories
- Working with dates, timestamps and data types
- Data validation, logging and error handling basics

DETAILED CLASS PLAN

#### Class 1: Data Engineering Roles, Lifecycle and Architecture

- Responsibilities of Data Engineers and downstream consumers
- Data lifecycle, source systems and pipeline stages
- Batch vs streaming; ETL vs ELT

✓ Hands-on lab: Draw a simple end-to-end pipeline for an e-commerce order dataset.

#### Class 2: Python Refresher for Data Work

- Functions, collections, modules and exceptions
- File paths, reading/writing files and data types
- Working with timestamps and structured records

✓ Hands-on lab: Write a Python script that reads a CSV file, validates required columns and writes a cleaned output.

### Class 3: JSON, APIs and Project 1 Start

- JSON structures and nested data
- REST API requests, pagination concepts and response handling
- Logging, retries and safe configuration

✓ Hands-on lab: Project 1: Python Data Ingestion - ingest sample CSV/JSON or public demo API data, validate it and save a standardised output.

## WEEK 02 — Advanced SQL and Data Modelling

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

### Module 3: Intermediate and Advanced SQL

- Joins: INNER, LEFT, RIGHT and FULL OUTER concepts
- Window functions: ROW_NUMBER, RANK, DENSE_RANK and LAG/LEAD
- Aggregations, grouping sets concepts and analytical queries
- Query execution plans and introductory performance tuning
- Subqueries and Common Table Expressions (CTEs)
- CASE expressions, date functions and string functions
- NULL handling, duplicate detection and data reconciliation

### Module 4: Data Modelling and Database Design

- OLTP vs OLAP systems
- Fact tables, dimension tables and measures
- Surrogate keys, natural keys and slowly changing dimensions (SCD) concepts
- Choosing grain and designing an analytical model
- Normalisation fundamentals and denormalisation trade-offs
- Star schema and snowflake schema
- Primary keys, foreign keys and referential integrity

DETAILED CLASS PLAN

### Class 4: Advanced SQL for Data Engineering

- CTEs, subqueries and multi-table joins
- Window functions and ranking
- Date handling, deduplication and data quality queries

✓ Hands-on lab: Write analytical SQL queries against a retail or finance dataset.

### Class 5: SQL Performance and Data Validation

- Indexes and query-plan intuition
- Filter and join optimisation concepts
- Reconciliation, duplicate and NULL checks

✓ Hands-on lab: Compare alternative query approaches and build validation queries.

### Class 6: Data Modelling and Project 2

- OLTP vs OLAP and analytical modelling
- Fact/dimension tables, grain and star schema
- SCD concepts and key design

✓ Hands-on lab: Project 2: Build an Analytical Data Model - design a star schema and create SQL transformations and validation queries.

## WEEK 03 — Data Ingestion, ETL/ELT and Pipeline Development

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

### Module 5: Data Sources and Ingestion Patterns

- Common data sources: relational databases, files, APIs and event feeds
- Change Data Capture (CDC) concept and use cases
- Handling pagination, rate limits and retries for APIs
- Landing/raw zones and ingestion metadata
- Full load vs incremental load
- Watermarks, timestamps and high-watermark patterns
- Schema drift and evolving source structures

## Module 6: ETL/ELT Pipeline Design and Transformation

- ETL vs ELT implementation patterns
- Data cleansing, standardisation and type casting
- Error handling, dead-letter/quarantine concepts
- Reusable Python pipeline components and basic testing
- Extract, validate, transform and load stages
- Idempotency and safe reruns
- Pipeline configuration and environment separation

DETAILED CLASS PLAN

### Class 7: Ingestion Patterns and Incremental Loads

- File, database and API ingestion
- Full vs incremental extraction
- Watermarks and CDC concepts

✓ Hands-on lab: Build a small incremental ingestion example using a timestamp or ID watermark.

### Class 8: ETL/ELT Transformation Workflow

- Extract, transform and load stages
- Cleaning, standardisation and business rules
- Idempotency, retries and quarantine records

✓ Hands-on lab: Transform raw customer/order data into curated tables.

### Class 9: Pipeline Testing and Project 2 Integration

- Schema checks and row-count reconciliation
- Reusable pipeline structure and configuration
- Document pipeline stages and failure cases

✓ Hands-on lab: Extend Project 2 with a repeatable ingestion-to-curated-data workflow.

## WEEK 04 — Data Lakes, Data Warehouses and Lakehouse Architecture

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

CareerByteCode
Learning Made Simple

## Module 7: Data Lake and Data Warehouse Fundamentals

- Data Lake purpose and architecture
- Object storage and file formats: CSV, JSON and Parquet
- Data Warehouse concepts and analytical workloads
- Data Lake vs Data Warehouse: strengths and trade-offs
- Raw, cleaned and curated data zones
- Partitioning and file organisation
- Warehouse loading patterns and dimensional models

## Module 8: Lakehouse and Modern Data Platform Concepts

- Why Lakehouse architecture emerged
- Delta Lake fundamentals and table versioning concepts
- Metadata/catalogue and data discovery concepts
- Choosing architecture for analytics and AI workloads
- Open table format concepts and ACID-style reliability
- Schema enforcement and schema evolution
- Medallion architecture: Bronze, Silver and Gold

DETAILED CLASS PLAN

### Class 10: Data Lake and File Storage Design

- Object storage, zones and folder organisation
- CSV, JSON and Parquet comparison
- Partitioning and file-size considerations

✓ Hands-on lab: Design a raw/clean/curated data lake layout for a healthcare or telecom use case.

### Class 11: Data Warehouse and Dimensional Architecture

- Fact and dimension tables
- Star schema and warehouse loading flow
- Analytical query patterns

✓ Hands-on lab: Map source tables into a simple warehouse model.

### Class 12: Lakehouse and Project 3 Preparation

- Lakehouse concepts, Delta Lake and ACID overview
- Bronze/Silver/Gold medallion layers
- Schema evolution and table reliability

✓ Hands-on lab: Prepare a Lakehouse architecture and transformation plan for Project 3.

## WEEK 05 — Apache Spark and PySpark

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

### Module 9: Apache Spark Fundamentals

- Why distributed processing is needed
- SparkSession and DataFrame API
- Partitions, shuffles and execution concepts
- Spark SQL and basic aggregations
- Spark architecture: driver, executors and cluster manager concepts
- Transformations vs actions and lazy evaluation
- Reading and writing CSV, JSON and Parquet

### Module 10: PySpark Data Transformation

- DataFrame selection, filtering and column expressions
- Handling nulls, duplicates and data types
- User-defined functions: when to avoid and alternatives
- Writing curated Parquet/Delta-style outputs where supported
- Joins, groupBy and aggregations
- Window functions in PySpark
- Partitioning and performance fundamentals

DETAILED CLASS PLAN

### Class 13: Spark Architecture and DataFrames

- Distributed processing concepts and Spark components
- SparkSession, DataFrames and lazy evaluation
- Read data and inspect schemas

✓ Hands-on lab: Load a dataset in PySpark and perform basic transformations.

### Class 14: PySpark Transformations and Joins

- Filter, select, withColumn and aggregations
- Joins and handling duplicate columns
- Null handling and type casting

✓ Hands-on lab: Transform multiple industry datasets into a unified DataFrame.

### Class 15: Spark SQL, Performance and Project 3

- Spark SQL and window functions
- Partitions, shuffles and caching concepts
- Write partitioned output in Parquet

✓ Hands-on lab: Project 3: PySpark Data Processing - process multi-file or multi-source data, transform it into curated outputs and document the pipeline.

## WEEK 06 — Apache Airflow and Workflow Orchestration

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

### Module 11: Airflow Fundamentals and DAG Design

- Why orchestration is needed
- DAGs, tasks, operators and dependencies
- Task dependencies and branching
- Running and inspecting DAGs in a local or training environment
- Airflow architecture and core components
- Scheduling, start dates and catch-up concepts
- Connections, variables and configuration

## Module 12: Reliable Orchestration and Pipeline Operations

- Retries, timeouts, failure handling and alerts
- Task logging and debugging
- Parameterised and reusable DAG patterns
- Pipeline observability and operational practices
- Backfills and reruns
- Sensors and waiting for upstream data
- Triggering Python, SQL and Spark jobs

DETAILED CLASS PLAN

### Class 16: Airflow Architecture and First DAG

- Scheduler, webserver/API, executor and metadata database concepts
- DAGs, tasks, operators and dependencies
- Create and run a simple DAG

✓ Hands-on lab: Build a DAG with Python tasks and visible dependencies.

### Class 17: Scheduling, Retries and Failure Handling

- Schedules, retries, timeouts and backfills
- Connections, variables and secrets/configuration
- Logs, debugging and alerting concepts

✓ Hands-on lab: Simulate a task failure and configure retry and recovery behaviour.

### Class 18: Airflow with Data Pipelines

- Orchestrate Python ingestion and SQL transformation tasks
- Trigger or coordinate a Spark job conceptually/in a lab
- DAG design patterns and Project 4 planning

✓ Hands-on lab: Create a scheduled workflow that runs ingestion, validation and transformation tasks.

## WEEK 07 — Kafka Fundamentals and Cloud Data Engineering

2 MODULES | 3 CLASSES | 6 INSTRUCTOR-LED HOURS

CareerByteCode
Learning Made Simple

## Module 13: Apache Kafka and Streaming Fundamentals

- Why event streaming is used
- Producers and consumers
- Message keys, ordering and retention concepts
- Batch vs streaming and when to use each
- Kafka architecture: brokers, topics and partitions
- Consumer groups and offsets
- Basic Python producer/consumer demonstration

## Module 14: Cloud Data Engineering with One Selected Provider

- Cloud data engineering architecture and shared responsibility
- Identity and access management for data services
- Cloud data warehouse or analytics service overview
- Connect cloud storage/processing concepts to the course pipeline
- Cloud object storage and data lake foundations
- Managed ingestion, ETL and data-processing service overview
- Networking, encryption, logging and cost-awareness fundamentals

DETAILED CLASS PLAN

### Class 19: Kafka Concepts and Basic Producer/Consumer

- Topics, brokers, partitions and messages
- Producer, consumer, consumer groups and offsets
- Basic event flow and streaming use cases

✓ Hands-on lab: Run a small local Kafka demonstration or instructor-guided producer/consumer exercise.

### Class 20: Selected Cloud Provider: Storage, Identity and Data Services

- AWS, Azure or GCP account and resource structure
- Cloud object storage and least-privilege access
- Overview of the provider's managed data ingestion/processing and warehouse services

✓ Hands-on lab: Upload and organise sample data in a sandbox cloud storage service and review access controls.

### Class 21: Cloud Pipeline Integration and Project 4 Build

- Connect local or managed processing to cloud storage
- Cloud pipeline architecture and service selection
- Logging, permissions, cost controls and cleanup

✓ Hands-on lab: Start Project 4: Cloud Data Pipeline - design and implement a guided ingestion-to-storage-to-processing workflow using the selected provider.

## WEEK 08 — Data Quality, Monitoring and End-to-End Capstone

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

### Module 15: Data Quality, Monitoring and Reliability

- Data quality dimensions: completeness, accuracy, validity, consistency and timeliness
- Row counts, null checks, uniqueness and referential checks
- Freshness, latency and volume checks
- Documentation, lineage and operational handover
- Schema validation and data contracts concepts
- Logging, metrics and pipeline status monitoring
- Failure recovery, reruns and alerting

### Module 16: End-to-End Data Engineering Capstone and Career Readiness

- Define a business problem and data sources
- Build Python/SQL/Spark processing steps
- Use Kafka fundamentals where relevant as an optional extension
- Validate outputs, document architecture and present results
- Design ingestion, raw, transformed and serving layers
- Orchestrate the workflow with Airflow
- Use the selected cloud provider for storage or processing
- Resume, portfolio and interview preparation

DETAILED CLASS PLAN

### Class 22: Data Quality and Pipeline Reliability

- Validation rules and data quality dimensions
- Monitoring, freshness and failure detection
- Retries, recovery and operational documentation

✓ Hands-on lab: Add quality checks and logging to an existing pipeline.

### Class 23: Capstone Integration and Validation

- Connect ingestion, transformation, storage and orchestration
- Run end-to-end pipeline and inspect outputs
- Troubleshoot failures and verify data quality

✓ Hands-on lab: Complete the final cloud pipeline, test a successful run and demonstrate a controlled failure/recovery.

### Class 24: Capstone Presentation and Career Preparation

- Present architecture and data flow
- Explain design decisions, tools and trade-offs
- Walk through code, DAGs, data model and validation
- Resume project bullets, portfolio and mock interview

✓ Hands-on lab: Present Project 4 and receive instructor feedback.

# Four Major Hands-On Projects

### P1 Python-Based Data Ingestion Pipeline

**Schedule:** Week 1

Ingest structured and semi-structured data from CSV, JSON and/or a public demo API. Validate required fields, handle malformed records, standardise data types and save the output in a raw/landing location.

**Tools:** Python, requests or equivalent HTTP library, CSV/JSON, logging

### P2 SQL Data Warehouse and Transformation Model

**Schedule:** Weeks 2-3

Design a simple star schema for a finance, retail or e-commerce scenario. Load source data, write SQL transformations, use joins and window functions, validate results and document the analytical model.

**Tools:** SQL database, SQL scripts, ER diagram or schema documentation

### P3 PySpark Lakehouse Processing Pipeline

**Schedule:** Weeks 4-5

Process multiple raw files using PySpark. Implement Bronze/Silver/Gold-style transformations, handle duplicates and nulls, join datasets, write Parquet outputs and document partitioning and data-quality checks.

**Tools:** Apache Spark, PySpark, Parquet, Delta Lake concepts where supported

### P4 Orchestrated Cloud Data Engineering Pipeline

**Schedule:** Weeks 6-8

Build an end-to-end pipeline that ingests data, validates it, transforms it with Python/SQL/PySpark, orchestrates steps with Airflow and stores outputs in the selected cloud provider. Include logging, retries, data-quality checks and a final architecture presentation. Kafka can be demonstrated as an optional event-ingestion extension.

**Tools:** Apache Airflow, Python, SQL, Spark/PySpark, one selected cloud provider

CareerByteCode

## Tools and Technologies

DATAFOUNDRY DATA ENGINEERING

| AREA | TOOLS / TECHNOLOGIES |
|---|---|
| **Programming and data formats** | Python, JSON, CSV, Parquet |
| **Database and SQL** | PostgreSQL or another training relational database; SQL |
| **Data processing** | Apache Spark, PySpark, Spark SQL |
| **Workflow orchestration** | Apache Airflow |
| **Streaming fundamentals** | Apache Kafka |
| **Data architecture** | Data Lake, Data Warehouse, Lakehouse, Bronze/Silver/Gold concepts |
| **Cloud - one provider per batch** | AWS, Azure or Google Cloud; selected storage and managed data services |
| **Development environment** | VS Code or equivalent, Jupyter/Notebooks, Git |

| AREA | TOOLS / TECHNOLOGIES |
|---|---|
| **Quality and operations** | Python logging, SQL validation queries, schema and data-quality checks |

Cloud service mapping will be finalised for each batch. Examples include AWS S3 and Glue/Redshift, Azure Data Lake Storage and Data Factory/Fabric services, or Google Cloud Storage and Dataflow/BigQuery. These are alternative provider-specific examples, not a promise to teach all three cloud stacks in one batch.

## Assessment and Evaluation

DATAFOUNDRY DATA ENGINEERING

| ASSESSMENT | WEIGHT | EVALUATION FOCUS |
|---|---|---|
| **Class coding exercises and practical labs** | 20% | Python, SQL, ingestion, Spark and Airflow exercises. |
| **Weekly assignments and quizzes** | 15% | Data concepts, architecture, SQL, processing and orchestration knowledge. |
| **Projects 1-3** | 30% | Correct ingestion, modelling, transformations, data quality and documentation. |
| **Final Project 4 capstone** | 25% | End-to-end pipeline, cloud integration, orchestration, reliability and presentation. |
| **Participation and technical communication** | 10% | Troubleshooting, collaboration, explanation of design decisions and documentation. |

**Suggested completion requirements:** learners should meet attendance requirement, submit all four projects and achieve the published passing threshold. CareerByteCode should define exact attendance and pass criteria before enrolment.

## Training Methodology and Delivery

DATAFOUNDRY DATA ENGINEERING

- Hybrid delivery with consistent class objectives, lab instructions and project rubrics for classroom and online learners.
- Recommended two-hour class rhythm: 20-30 minutes concept explanation, 35-45 minutes instructor demonstration, 35-45 minutes guided hands-on practice and 5-10 minutes recap.
- Use small datasets initially, then gradually introduce multi-file and multi-source datasets from finance, healthcare, retail, e-commerce and telecom scenarios.
- Provide starter repositories, notebooks, sample data, SQL scripts, Airflow DAG templates, architecture diagrams and project rubrics.
- Use synthetic, public or properly de-identified datasets. Do not use confidential customer, patient or employee data in training labs.
- Cloud labs should use sandbox accounts, least-privilege permissions, budget alerts and resource cleanup instructions.
- Provide catch-up material and a defined doubt-clearing process where operationally possible.

## Career Support and Portfolio

DATAFOUNDRY DATA ENGINEERING

- Course completion certificate subject to published attendance and assessment criteria.
- Resume guidance focused on Python, SQL, Spark, Airflow, cloud fundamentals and completed project work.
- Portfolio containing code repositories, SQL scripts, pipeline diagrams, Airflow DAGs, sample outputs and project documentation.
- Mock interviews covering Python data processing, SQL, ETL/ELT, Spark, Airflow, data modelling and cloud fundamentals.
- Scenario-based interview preparation covering failed pipelines, duplicate records, late-arriving data, schema changes and data-quality issues.
- Guidance on presenting project architecture, design choices, limitations and troubleshooting steps honestly.

# Commercial Details

DATAFOUNDRY DATA ENGINEERING

**Course fee:** ₹40,000 + applicable GST per learner.

**Duration:** 8 weeks | 24 classes | 48 instructor-led hours.

**Delivery:** Hybrid: classroom + online.

**Cloud delivery:** One provider - AWS, Azure or Google Cloud - will be selected and announced before each batch.

**Included:** Structured classes, guided labs, four project briefs, assessments, certificate subject to completion criteria, resume support, portfolio guidance, mock interviews and interview preparation.

**Cloud usage charges:** Any cloud consumption or paid service charges should be disclosed before the lab and managed through CareerByteCode sandbox accounts or separately approved learner accounts.

# Important Academic and Operational Notes

DATAFOUNDRY DATA ENGINEERING

- The 48 instructor-led hours are intensive. Learners should plan regular independent practice between sessions.
- Basic Python and SQL are prerequisites; the course is not designed to teach programming from zero.
- The selected cloud provider must be confirmed before the batch. The course fee does not imply full hands-on training across AWS, Azure and GCP.
- Kafka is a fundamentals module with a guided producer/consumer exercise; advanced streaming architecture is outside the core scope.
- Airflow and Spark labs may use local containers or a managed training environment depending on available hardware and cloud access.
- Cloud resources should be created with least privilege, cost controls and teardown instructions. Avoid leaving paid services running after labs.
- Use public, synthetic or de-identified datasets, especially for healthcare, finance and telecom exercises.

CareerByteCode
Learning Made simple

CareerByteCode
Learning Made simple

# THANK YOU

Join the next batch, learn from practitioners, and leave with projects you can show and explain.

**CareerByteCode Admissions** · support@careerbytecode.in · www.careerbytecode.in/courses/datafoundry-data-engineering/

CareerByteCode is a brand of Cloudnloud Technologies Pvt Ltd, Chennai. CareerByteCode Team.