Day 1: Python Essentials for Dask and Data Analysis

Session 1: Introduction to Python for Data Analysis (2 Hours)

- **Key Python Concepts for Data Analysis**:
    - Data structures: Lists, dictionaries, and tuples.
    - Comprehensions and generators.
- **Overview of Python Libraries for Data Analysis**:
    - NumPy, Pandas, and SciPy overview.
    - Introduction to Matplotlib for visualization.

Hands-On Lab:

- Perform data wrangling using Pandas.
- Calculate basic statistics using NumPy.

Session 2: Setting Up the Environment for Python and Dask (2 Hours)

- **Installing Python and Essential Libraries**:
    - Setting up a virtual environment using `venv` or `conda`.
    - Installing Dask and related libraries.
- **Introduction to Jupyter Notebooks**:
    - Setting up JupyterLab for interactive data exploration.

Hands-On Lab:

- Install and configure Python, Dask, and JupyterLab.

Session 3: Introduction to Dask and Parallel Computing (4 Hours)

- **Overview of Dask Features and Advantages**:
    - Differences between Dask and traditional Python tools.
    - Use cases for Dask in parallel computing.
- **Introduction to Parallel Computing in Python**:
    - Threads, processes, and distributed systems.
    - Role of Dask schedulers.

Hands-On Lab:

- Perform parallel computation using Dask Bag.
- Compare execution times between pure Python and Dask.

Day 2: Scaling Python Libraries with Dask

Session 1: Scaling NumPy with Dask Arrays (2 Hours)

- **Dask Arrays**:
    - Chunks and blocked algorithms.
    - Overlapping computations.
- **SciPy Integration with Dask**:
    - Using Dask with `scipy.stats` and `LinearOperator`.

Hands-On Lab:

- Perform large-scale computations using Dask Arrays.
- Apply statistical operations on large datasets with Dask and SciPy.

Session 2: Scaling Pandas with Dask DataFrames (2 Hours)

- **Introduction to Dask DataFrames**:
    - How Dask extends Pandas functionality.
    - Partitioning and distributed data structures.
- **Working with Large Datasets**:
    - Aggregations and group-by operations.
    - Joins, merges, and filtering with Dask.

Hands-On Lab:

- Process a multi-gigabyte CSV file using Dask DataFrames.

Session 3: Analyzing Performance and Graph Computation (4 Hours)

- **Scheduler and Diagnostics**:
    - Overview of Dask's schedulers (single-threaded, threaded, distributed).
    - Monitoring performance with Dask's diagnostics tools.
- **Graph Computation in Dask**:
    - Building and visualizing Dask task graphs.
    - Optimizing graph computations.

Hands-On Lab:

- Visualize task graphs using the Dask dashboard.

● Analyze performance bottlenecks in parallel computations.

Day 3: Optimizing and Deploying Dask

Session 1: Optimizing Dask Workflows (2 Hours)

- **Best Practices for High Performance**:
  - Chunk sizing and memory optimization.
  - Avoiding common pitfalls in Dask workflows.
- **Debugging Parallel Programs**:
  - Tools and techniques for debugging Dask applications.

Hands-On Lab:

- Optimize a slow-running Dask pipeline.
- Debug a Dask-based data pipeline with errors.

Session 2: Deploying Dask Clusters (2 Hours)

- **Cluster Deployment Options**:
  - Local clusters and distributed setups.
  - Deploying Dask on Kubernetes and cloud platforms.
- **Working with GPUs**:
  - GPU acceleration with Dask-cuDF and Dask-ML.

Hands-On Lab:

- Deploy a simple Dask cluster locally.
- Set up a basic Dask cluster on Kubernetes.

Session 3: Advanced Use Cases and Integrations (4 Hours)

- **Connecting to Remote Data Sources**:
  - Working with data stored on S3, Azure Blob, or HDFS.
  - Loading and processing data directly from cloud storage.
- **Outbound Delivery**:
  - Exporting results to databases or cloud storage.
  - Integrating Dask with visualization tools like Matplotlib and Bokeh.

Hands-On Lab:

- Process remote data stored on AWS S3.
- Visualize results with Matplotlib and Dask.

Day 4: Real-World Scenarios and Case Studies

Session 1: End-to-End Data Pipeline Creation (4 Hours)

- **Building a Complete Data Pipeline**:
  - Ingesting raw data.
  - Transforming and analyzing data using Dask.
  - Exporting results for reporting or further analysis.
- **Case Study: Large-Scale Financial Data Analysis**:
  - Process and analyze stock market data using Dask.

Hands-On Lab:

- Implement an end-to-end data pipeline with Dask Arrays and DataFrames.

Session 2: Integration and Monitoring (4 Hours)

- **Integrating Dask into Production Workflows**:
  - Using Dask with workflow managers (e.g., Apache Airflow).
  - Monitoring distributed clusters in production.
- **Real-World Challenges and Solutions**:
  - Managing memory and handling failures.
  - Strategies for scaling workflows effectively.

Hands-On Lab:

- Monitor a Dask workflow using the dashboard.
- Integrate Dask with Apache Airflow for orchestration.