- Introduction
  - Describe what makes data “Big Data”
  - List data types stored and analyzed in Hadoop
  - Compare Hadoop vs traditional systems
  - Describe how Big Data and Hadoop fit into your current infrastructure and environment
  - Fundamentals of:
    - the Hadoop Distributed File System (HDFS)
    - YARN
    - MapReduce
    - Recognize use cases for Hadoop
    - Describe the business value of Hadoop
  - Understanding HDFS Architecture
    - Hadoop Master-Slave Architecture
    - NameNode, DataNode, Secondary Node
    - Learn about JobTracker, TaskTracker
  - Need for Hadoop frameworks:
    - Pig, Hive, HCatalog, Storm, Solr, Spark, HBase, Oozie, Ambari, ZooKeeper, Sqoop, Flume, and Falcon
    - Describe new technologies like Tez and the Knox Gateway

- Hadoop Configuration
  - Hadoop Modes
  - Installation of Hadoop in LocalMode, Pseudo Dist modes
  - Hadoop Terminal Commands
  - Cluster Configuration
  - Web Ports
  - Hadoop Configuration Files
  - Reporting, Recovery
  - MapReduce in Action

- Understanding Hadoop MapReduce Framework
  - Overview of the MapReduce Framework
  - Use cases of MapReduce
  - MapReduce Architecture
  - Anatomy of MapReduce Program
  - Mapper/Reducer Class, Driver code
  - Understand Combiner and Partitioner

- Hadoop 2.0, YARN, MRv2
  - Hadoop 1.0 Limitations
  - MapReduce Limitations
  - HDFS 2: Architecture
  - HDFS 2: High availability
  - HDFS 2: Federation
  - YARN Architecture

  - Classic vs YARN
  - YARN multitenancy
  - YARN Capacity Scheduler

- Advanced MapReduce - Part 1
  - Write your own Partitioner
  - Writing Map and Reduce in Python
  - Map side/Reduce side Join
  - Distributed Join
  - Distributed Cache
  - Counters
  - Joining Multiple datasets in MapReduce

- Advanced MapReduce - Part 2
  - MapReduce internals
  - Understanding Input Format
  - Custom Input Format
  - Using Writable and Comparable
  - Understanding Output Format
  - Sequence Files
  - JUnit and MRUnit Testing Frameworks
  - Debugging an MR Job
  - Serialization formats
    - Avro/Protobuf/Thrift
  - Compression

- Apache Pig
  - Pig Installation
  - Pig Run modes
  - PIG vs MapReduce
  - PIG Architecture & Data types
  - PIG Latin Relational Operators
  - PIG Latin Join and CoGroup
  - PIG Latin Group and Union
  - Describe, Explain, Illustrate
  - PIG Latin: File Loaders & UDF

- **Mini Hackathon -1**
  - Use case would be provided a day before on the problem that need to be solved

- Apache Hive and HiveQL
  - What is Hive
  - Hive Installation and Run modes
  - Hive DDL - Create/Show Database
  - Hive DDL - Create/Show/Drop Tables
  - Hive DML - Load Files & Insert Data
  - Hive SQL - Select, Filter, Join, Group By
  - Hive Architecture & Components
  - Difference between Hive and RDBMS

- Advance HiveQL
  - Multi-Table Inserts
  - Joins
  - Grouping Sets, Cubes, Rollups
  - Custom Map and Reduce scripts
  - Hive SerDe
  - Hive UDF
  - Hive UDAF

- HCatalog
  - Installation
  - Uses and configuration

- Apache Flume, Sqoop
  - Installation of Sqoop
  - Sqoop - How Sqoop works
  - Sqoop Architecture
  - Installation of Flume
  - Compare and contrast with other data transport frameworks
  - Flume - How it works
  - Flume Complex Flow - Multiplexing

- NoSQL Databases
  - CAP theorem
  - ACID v/s BASE
  - RDBMS vs NoSQL
  - Key Value stores: Memcached, Riak
  - Key Value stores: Aerospike, Redis, Dynamo DB
  - Column Family: Cassandra, HBase
  - Graph Store: Neo4J
  - Document Store: MongoDB, CouchDB

- Apache HBase
  - When/Why to use HBase
  - HBase Architecture/Storage
  - HBase Installation and Configuration
  - HBase Data Model
  - HBase Families/ Column Families
  - HBase Master
  - HBase vs RDBMS
  - Access HBase Data
  - Monitoring and managing HBase
  - How Apache Phoenix works with HBase

- **Mini Hackathon -2**
  - Use case would be provided a day before on the problem that need to be solved

- Apache Zookeeper

  - What is Zookeeper
  - Zookeeper Data Model
  - Installing and Configuring
  - ZNode Types
  - Sequential ZNodes
  - Running Zookeeper
  - Zookeeper use cases
  - How HBase integrates with ZooKeeper
  - Curator frameworks

- Apache Kafka
  - What is Kafka
  - Compare and contrast with other messaging systems
  - Use cases of Kafka
  - Kafka Broker, Producer and Consumers
  - Writing a high level producer, consumer
  - Kestrel introduction

- CDH Introduction
  - Components of CDH
  - Using the VM
  - Hue Interface

- Apache Oozie
  - Oozie installation
  - Oozie - Simple/Complex Flow
  - Oozie Service/ Scheduler
  - Use Cases - Time and Data triggers
  - Other worflow engines, Falcon, Azkaban

- Apache Drill
  - Drill Installation
  - Drill Architecture and Usecases
  - Using Tableau with Drill

- Impala
  - Impala Architecture and Usecases
  - Using Excel/QlikView with Impala

- Storm and Trident
  - Recognize differences between batch and real-time data processing
  - Define Storm elements including tuples, streams, spouts,topologies, worker processes, executors, and stream groupings
  - Explain and install Storm architectural components including Nimbus, Supervisors, and ZooKeeper cluster

- Big Data Analytics
  - Text Analytics Essentials
  - Introduction to Solr
  - Introduction to Jaql

  - Elasticsearch

- Big Data in the Cloud
  - Amazon Web Services
  - Concepts: Pay pay use model
  - Amazon S3, EC2, EMR
  - Google Cloud Platform
  - Google Big Query

- Lambda Architecture
  - Concept
  - Hadoop + Stream processing integration
  - Architecture examples

- Data Mining with Mahout
  - Clustering
  - Classification
  - Batch-based collaborative filtering

- **Main Hackathon**
  - Use case would be provided a day before on the problem that need to be solved

- Real time project