

## DVS SPARK Course Content

### INTRODUCTION AND OVERVIEW OF APACHE SPARK

#### SCALA Programming language

##### 1. Scala introduction

- ✓ It's Functional programming(FP)
- ✓ Also Object-Oriented Programming(OOPs)
- ✓ After Scala, a simple definition for FP and OOPs
- ✓ Where are all Scala is using
- ✓ History of Scala
- ✓ Scala program flow
- ✓ First Scala program
- ✓ Immutability
- ✓ Interactive shell REPL

##### 2. Variables

- ✓ variables
- ✓ Properties of variable
- ✓ Creating variable
- ✓ val keyword
- ✓ var keyword
- ✓ Summary: when should we go for val and var
- ✓ Expressions

##### 3. Data types

- ✓ Types of data types
  - Byte
  - Short
  - Int
  - Long
  - Float
  - Double
  - Char
  - Boolean

#### 4. Flow control

- ✓ if
- ✓ if else
- ✓ if else if
- ✓ while
- ✓ do while
- ✓ for
- ✓ switch and case
- ✓ pattern matching
- ✓ return
- ✓ break

#### 5. Scala is functional programming

- ✓ What is a function?
- ✓ Why function?
- ✓ Where function is using?
- ✓ Functions are two types
  - Without parameters
  - With parameters
    - VarArg parameters
- ✓ Higher order function
- ✓ Pure functions
- ✓ Examples on functions

#### 6. Scala is Object Oriented Programming too

- ✓ What is OOPs (Object Oriented Programming Principles)
  - class
  - object
  - Why should we create an object for a class?
    - What is an Object
    - Characteristics about object
- ✓ Data Hiding or Information Hiding:
- ✓ Abstraction
- ✓ Encapsulation:
- ✓ Methods
  - Types of methods
  - Zero parameterized methods
  - Parameterized methods

- ✓ Constructors in Scala
  - What is the purpose of constructor?
  - When constructor will get execute?
  - How many times constructor will get execute?
  - Does developer need to call constructor explicitly like a method?
  - Types of constructor
  - Without parameters Primary constructor
  - Primary constructor which are having parameters
  - Auxiliary Constructor
- ✓ Inheritance
  - What is inheritance?
  - How to implement inheritance?
  - Still expecting more explanation then...
  - Advantages of Inheritance:
  - Types of Inheritance
    - Single Inheritance
    - Multi-level Inheritance
    - Multiple Inheritance
    - Why multiple inheritances is not supporting?
- ✓ Polymorphism
  - What is polymorphism
  - Dynamic Polymorphism
  - Method Overloading
  - Cases in overloading
    - Difference in the number of parameters.
    - Difference in the datatype of parameters.
    - Difference in the order or sequence of parameters.
- ✓ Can we overload main() method?
- ✓ Method overriding
  - When should we go for overriding?
  - Difference between Method overloading and Method overriding
- ✓ final keyword
  - final method
  - final class
  - Smart question: If we are using final keyword then are, we missing OOPs features?
- ✓ Abstract class
  - Abstract keyword
  - Types of methods
    - Implemented method

- Unimplemented method
- ✓ Abstract method
- ✓ Abstract class
- ✓ Abstract variable
- ✓ If you have time
  - Please prepare given scenarios
- ✓ trait
  - trait keyword
  - What is trait?
  - A single class can extends multiple traits
  - If you have time
    - Please prepare given scenarios
- ✓ Different type of classes
  - Normal
  - Singleton
  - Standalone
- ✓ Singleton object
  - Purpose of singleton object
  - Difference between instance variable and singleton variable
  - How to access singleton variable
- ✓ Companion object
  - What is companion object
  - Advantage
  - Rules to define companion object
- ✓ Case class
  - Case keyword
  - Why case class?
  - Advantage
  - Difference between case class and normal class
- ✓ Implicites

## Spark Index

### 1. Spark Introduction

- ✓ What is a spark?
- ✓ Prerequisites to learn spark
- ✓ Purpose of Spark
- ✓ Spark is written in which programming language
- ✓ Can Spark integrate with Hadoop?
- ✓ What kind of files sparks support?
- ✓ Is Spark depending on Hadoop?
- ✓ History of Spark
- ✓ Spark features
- ✓ Introduction to spark Scala's and python shells
- ✓ How to deploy spark applications
  - Spark Deployment modes and their usage patterns
- ✓ Spark Architecture
  - Standalone cluster mode
  - Spark on YARN mode
- ✓ Apache Spark Components or modules
  - Core
  - SQL
  - Streaming
  - MLlib
  - GraphX
  - SparkR
- ✓ Cluster Managers
- ✓ Storage Layers for Spark
- ✓ Spark Execution Model
- ✓ Spark Terminology table
- ✓ Spark follows...
- ✓ Driver program
- ✓ Executors
- ✓ SparkContext
  - How many SparkContext objects can create for one application?
  - Stopping SparkContext object
  - SparkContext responsibilities
- ✓ Spark 1.x version
- ✓ Solution in Spark 2.x
- ✓ RDD

## 2. Programming with RDD (Spark's Data Abstraction)

- ✓ Importance of RDD
- ✓ Partitions in RDD
- ✓ Different ways of creating an RDDs
- ✓ RDD Lineage and Persistence
- ✓ RDD Partitioning & How It Helps Achieve Parallelization
- ✓ Caching
- ✓ Persistent
- ✓ Fault-Recovery Mechanism
- ✓ If RAM is inefficient to store RDD then where it stores?
- ✓ RDD features
- ✓ Spark RDD Operations
- ✓ Transformations
  - Types of Transformations
  - Narrow Transformations
  - Wide Transformations
- ✓ Actions
- ✓ Limitation of RDD
- ✓ RDD Operations
- ✓ Transformations & Actions
- ✓ Programs
- ✓ Coalesce and Repartition
- ✓ RDD Partitioning & How It Helps Achieve Parallelization
- ✓ Data Loading and Saving through RDDs
- ✓ Performing data transformations and aggregations/joins through RDDs
- ✓ RDD Advanced concepts – Accumulators, Broadcast variables
- ✓ Internals of Job execution in Spark

## 3. Spark SQL and DataFrames

- ✓ Need for Spark SQL, Spark SQL and its features
- ✓ Spark SQL Architecture
- ✓ Data Frames – A Spark SQL data abstraction
- ✓ Connecting to diverse Data Sources (HDFS, Hive, S3, RDBMS and NoSQL etc.)
- ✓ Loading and writing to different file formats (CSV, XML, JSON, Parquet, ORC)
- ✓ Interoperating with RDDs
- ✓ Building ETL pipelines through Spark SQL
- ✓ Data transformation and aggregation/joins through Spark SQL
- ✓ Spark SQL User Defined Functions (UDFs)
- ✓ Spark SQL integration with Hive – Loading and writing to Hive tables in a partitioned and bucketed manner
- ✓ Real time challenges and case studies

#### **4. Spark Streaming**

- ✓ Batch vs Streaming, Spark Streaming and its features
- ✓ Architecture and Abstraction
- ✓ DStreams, DStreams vs RDD
- ✓ Spark Streaming workflow, DStream Transformations
- ✓ Input Streams (Socket, HDFS, Twitter, Kafka)
- ✓ Kafka and its architecture
- ✓ Using Kafka as source in Spark Streaming
- ✓ Fault tolerance through Check pointing, Persist and Caching
- ✓ Batch and Window Sizes
- ✓ Aggregations through Stateful operators

#### **5. Upgrading Spark – Spark 2.x, 3.x**

- ✓ Advancements in Spark 2.x and 3.x
- ✓ New robust data abstraction – Dataset
- ✓ Unified entry point to all Spark libraries
- ✓ Introduction to Spark Structured Streaming
- ✓ Examples running through Spark Structured Streaming

#### **6. Overview sessions on Kafka, Cassandra**

#### **7. Project in Spark Streaming with Kafka and Cassandra**

#### **8. Spark advanced topics too**

#### **9. PYSPARK workshops once in 2-3 months**