SCALA and SPARK PROGRAMMING: -

Scala and Spark Training Prerequisites:

Attendees should be Java developers planning to develop Scala applications.

Hadoop Map Reducing, Hadoop Architecture.

Hands-on/Lecture Ratio

This course is 60% hands-on, 40% discussion, with the longest discussion segments lasting 20 minutes.

Software Needed on Each Student PC

- Java Runtime Environment 1.8.x
- Scala 2.12.x (Latest)
- https://www.scala-lang.org/download/
- Scala IDE for Eclipse is recommended;
- Spark 2.1.x
- http://spark.apache.org/downloads.html
- Hadoop 2.7.x
- Cloudera Vmware (Free Downloadable)
- Vmware Workstation / Fusion / Virtual Box

Scala Training Objectives

- Program in Scala
- Understand Scala's approach to object-orientation
- Master the use of functional programming techniques in Scala
- Understand how to perform TDD (test-driven development) using Scala

Saprk Training Objectives

- Install Spark and implement Spark operations on Spark Shell
- Understand the role of Spark RDD
- Implement Spark applications on YARN (Hadoop)
- Learn Spark Streaming API

Scala Training Outline

Key Features of the Scala Language

- Everything is an object
- Class declarations
- Data typing
- Operators and methods
- Pattern matching
- Functions
- Anonymous and nested functions
- Traits

Basic Programming in Scala

- Built in types, literals and operators
- Testing for equality of state and reference
- Conditionals, simple matching and external iteration
- Working with lists, arrays, sets and maps
- Throwing and catching exceptions
- Adding annotations to your code
- Using standard Java libraries
- Using Scala with in java application and vice-versa

OO Development in Scala

- A minimal class declaration
- Understanding primary constructors
- Specifying alternative constructors
- Declaring and overriding methods
- Creating base classes and class hierarchies
- Creating traits and mixing them into classes
- How a Scala inheritance tree is linearised

Functional Programming in Scala

- Advanced uses of for expressions
- Understanding function values and closures
- Using closures to create internal iterators

Creating and using higher order functions
Practical examples of higher order functions
Currying and partially applied functions
Creating your own Domain Specific Languages(DSL's)

Exception handling in Scala

Try catch with case

Pattern Matching in Depth

Using the match keyword to return a value
Using case classes for pattern matching
Adding pattern guards to match conditions
Partially specifying matches with wildcards
Deep matching using case constructors
Matching against collections of items
Using extractors instead of case classes

Test Driven Development in Scala

Writing standard JUnit tests in Scala
Conventional TDD using the ScalaTest tool
Behaviour Driven Development using ScalaTest
Using functional concepts in TDD
Conclusion and Summary

Spark Training Outline

Introduction Big-data ecosystem

Hadoop Ecosystem Overview
HDFS Overview
YARN Overview
Spark Overview

Introduction to SPARK

Introduction to Spark Scala API
Hadoop Map Reducing Concept
Execute Apache Spark Tasks
Configuring Apache Spark
Building and Running Spark Applications
Resilient Distributed datasets
Hands On Session using SPARK API
Conclusion and Summary

Spark Basics

Using the Spark Shell
Introduction to RDDs (Resilient Distributed Datasets)
Functional Programming in Spark

Working with RDDs in Spark

Purpose and Structure of RDDs
Transformation, Actions and DAG
RDD Programming API
RDD Operations
Creating RDDs from a File (example)
Other General RDD Operations

Aggregating Data with Pair RDDs

Key-Value Pair RDDs
Map-Reduce
Other Pair RDD Operations

Spark RDD Persistence

RDD Lineage
RDD Persistence Overview
Distributed Persistence

Writing and Deploying Spark Applications

Spark Applications vs. Spark Shell
Creating the SparkContext
Building a Spark Application
Running a Spark Application
The Spark Application Web UI
Configuring Spark Properties
Logging