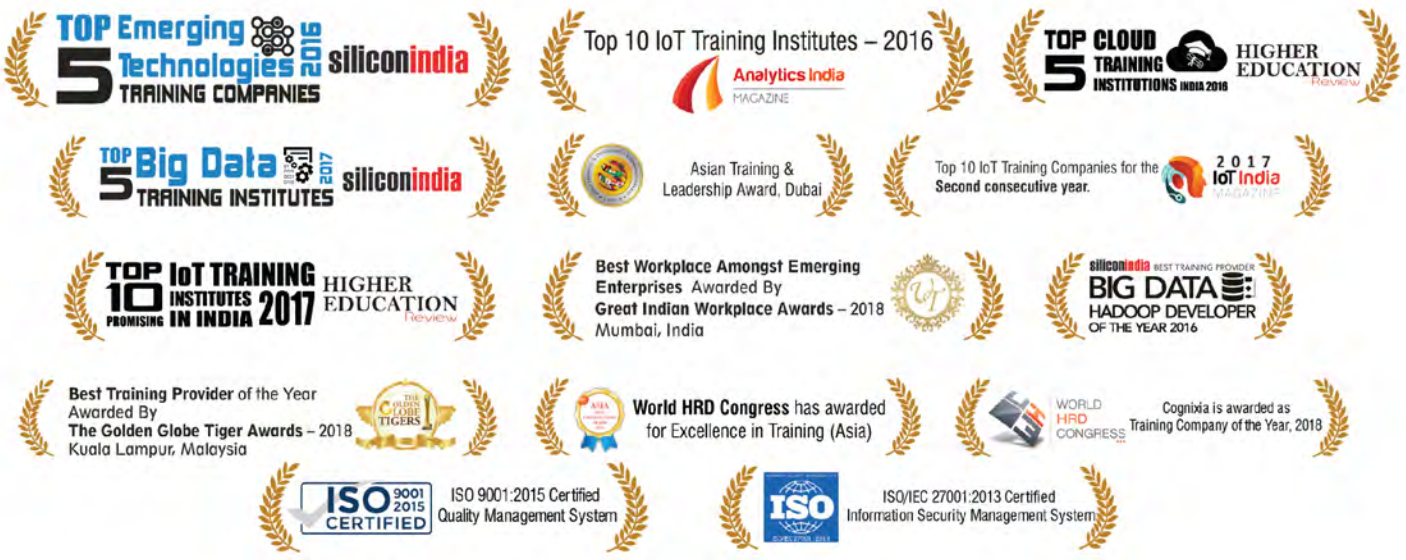
Cognixia

A Collabera LEARNING SOLUTIONS COMPANY

360°
Master's Program
Big Data Architect



join@cognixia.com www.cognixia.com

Cognixia
A Collabera LEARNING SOLUTIONS COMPANY

Big Data Architect

Program Modules

- Java
- Scala Programming
- Big Data Hadoop Developer
- Spark Development Training
- MongoDB

JAVA Training
Course Content

join@cognixia.com www.cognixia.com

Java Training

Topics covered in the training

Session Plan

Core Java Content :

1. Features of Java
2. Java Basics
3. Classes and Objects
4. Garbage Collection
5. Java Arrays
6. Referring Java Documentation
7. Wrapper classes
8. Inheritance
9. Polymorphism
10. Abstract Classes
11. Interfaces
12. Packages
13. Introduction to Exception Handling
14. Checked/Unchecked Exceptions
15. Using try, catch, finally, throw, throws
16. Exception Propagation
17. Pre-defined Exceptions
18. User Defined Exceptions
19. Overview of Java IO Package
20. Byte streams
21. Character streams
22. Object serialization & Object Externalization
23. Introduction to GUI Programming (swing)
24. Introduction to multithreading
25. Thread life cycle
26. Thread priorities
27. Using wait() & notify()
28. DeadLocks
29. JDBC Architecture
30. Using JDBCI API
31. Transaction Management

Java Training

Course Content - Servlets and JSP

Java Servlet Technology

- What is a Servlet?
- Servlet Life Cycle
- Initializing a Servlet
- Writing service methods
- Getting information from requests
- Constructing responses
- ServletContext and ServletConfig parameters
- Attributes- Context, Request and Session
- Maintaining Client State – Cookies/Url rewriting/Hidden Form Fields
- Session Management
- Servlet communication – include, forward, redirect
- WEB-INF and the Deployment Descriptor

Java Server Pages Technology

- What is a JSP page?
- The life cycle of a JSP page
- Execution of a JSP page
- Different types of tags (directive, standard actions, bean tags, expressions and declarative)
- Creating static content
- Creating dynamic content
- Using Implicit Objects within JSP Pages
- JSP Scripting Elements
- Including content in a JSP Page
- Transferring control to another web component – communication with servlet
- Param element
- JavaBeans component design conventions
- Why use a JavaBeans component?
- Creating and using a JavaBeans component
- Setting JavaBeans component properties
- Retrieving JavaBeans component properties
- Custom tags

Scala Programming Course Content

join@cognixia.com www.cognixia.com

Cognixia
A Collabera LEARNING SOLUTIONS COMPANY

Scala Programming

Topics covered in the training

PLAY FOR SCALA
Duration: - 2 Days

Scala Training Prerequisites
Attendees should be strong Java developers planning to develop Scala applications

Hands-on/Lecture Ratio
The course has 60% hands-on training and 40% discussion, with the longest discussion segments lasting about 20 minutes

Scala Training Materials
All students receive comprehensive courseware and a related textbook

Software participants need to have installed
- Java Runtime Environment 1.5 or later (1.6 or later recommended)
- Scala 2.8.x or later
- Scala IDE for Eclipse is recommended (Please reach out to us if you prefer using another Scala-aware IDE)

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
- Manipulate XML in Scala
- Write concurrent applications that are thread-safe

Scala Training Outline

Introduction to Scala

- Brief history of the Java platform
- Differences between the Java language and the platform
- Limitations of using Java for software development
- Possible features for an improved Java version
- How and why was Scala language created

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 code
- Using standard Java liberties
- Using Scala within Java application and vice-versa

OO development in Scala

- 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
- Linearizing a Scala inheritance tree

Functional programming in Scala

- Advanced uses of expressions
- Understanding function values & closures
- Creating and using higher order functions
- Creating and using higher order functions
- Practical examples of higher order functions
- Currying and partially applied functions
- Creating Domain Specific Languages (DSL)

Handling exceptions in Scala

- Try and Catch with Case

Pattern Matching

- 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
- Behavior Driven Development using ScalaTest
- Using functional concepts in TDD

XML Manipulating in Scala

- Using Scala to read and write xml using different parsers (Dom, Sax)
- Working with XML literals in code
- Embedding XPath like expressions
- Using Pattern Matching to process XML data
- Serializing and deserializing to and from XML
- Scala with database transaction

Writing Concurrent Apps

- Issues with conventional approaches to multi-threading
- Using actor-based approach to write thread-safe code
- Scala architecture for creating actor-based systems
- Different coding styles supported by the actor model

Scala web

- Scala with JAXB
- Scala to call/consume a REST/SOAP service
- Scala with logging information
- Using Scala in web application (JSP, Servlet)
- Conclusion

Introduction

- Introduction
- Module Outline
- What We Will Build
- History of Play!
- Philosophy
- Technologies
- Summary

Starting Up

- Introduction
- Downloading Play!
- The Play Command
- Compiling and Hot Deploy
- Testing
- IDE's
- Project Structure
- Configuration
- Error Handling
- Summary

Routing

- Introduction
- The Router
- Router Mechanics
- Routing Rules
- Play! Routes
- Play! Routes: HTTP Verbs
- Play! Routes: The Path
- Play! Routes: The Action Call
- Routing in Action
- Summary

Controllers, Actions, and Results

- Introduction
- Controllers
- Actions
- Results
- Session and Flash Scope
- Request Object
- Implementing the Contacts Stub Controller
- Summary

Views

- Introduction
- Play! Views
- Static Views
- Passing Arguments
- Iteration
- Conditionals
- Partials and Layouts
- Accessing the Session Object
- The Asset Route
- Summary

Data Access

- Introduction
- Agnostic Data Access
- The Domain Model
- Evolutions
- Finder and Listing Contacts
- The Form Object and Adding a Contact
- Editing a Contact
- Deleting a Contact
- Review
- Summary

The Global Object

- Introduction
- The Global Object
- Global Object Methods
- onStart
- onHandlerNotFound
- Summary

Big Data Hadoop Developer

join@cognixia.com www.cognixia.com

Cognixia
A Collabera LEARNING SOLUTIONS COMPANY

Big Data Hadoop Developer

Topics covered in the training

Introduction to Linux and Big Data Virtual Machine (VM)
Introduction/ Installation of VirtualBox and the Big Data VM Introduction to Linux - Why Linux? - Windows and the Linux equivalents - Different flavors of Linux - Unity Shell (Ubuntu UI) - Basic Linux Commands (enough to get started with Hadoop).

Understanding Big Data

- 3V (Volume- Variety- Velocity) characteristics
- Structured and Unstructured Data
- Application and use cases of Big Data limitations of traditional large Scale systems
- Distributed way of computing and its benefits (cost and scale)
- Opportunities and challenges with Big Data

HDFS (The Hadoop Distributed File System)

HDFS Overview and Architecture
- Deployment Architecture
- Name Node, Data Node and Checkpoint Node (aka Secondary Name Node)
- Safe mode
- Configuration files
- HDFS Data Flows (Read v/s Write)

How HDFS addresses fault tolerance?
- CRC Check Sum
- Data replication
- Rack awareness and Block placement policy
- Small files problem

HDFS Interfaces
- Command Line Interface
- File System
- Administrative
- Web Interface

Advanced HDFS features
- Load Balancer
- Dist Cp
- HDFS Federation
- HDFS High Availability
- Hadoop Archives

Map Reduce - 1 (Theoretical Concepts)
MapReduce overview

- Functional Programming paradigms
- How to think in a MapReduce way?

MapReduce Architecture
- Legacy MR v/s Next Generation MapReduce (aka YARN/ MRv2)
- Slots v/s Containers
- Schedulers
- Shuffling, Sorting
- Hadoop Data Types
- Input and Output Formats
- Input Splits - Partitioning (Hash Partitioner v/s Customer Partitioner)
- Configuration files
- Distributed Cache

MR Algorithm and Data Flow
- Word Count

Alternatives to MR - BSP (Bulk Synchronous Parallel)
- Adhoc querying
- Graph Computing Engines

Map Reduce - 2 (Practice)

Developing, debugging and deploying MR programs
- Stand alone mode (in Eclipse)
- Pseudo distributed mode (as in the Big Data VM)
- Fully distributed mode (as in Production)

MR API
- Old and the new MR API
- Java Client API
- Hadoop data types and custom Writable / WritableCom parables
- Different input and output formats
- Saving Binary Data using Sequence Files and Avro Files

Hadoop Streaming (developing and debugging non Java MR program s - Ruby and Python)

Optimization techniques

- Speculative execution

- Combiners
- JVM Reuse
- Compression

MR algorithm s (Non- graph)

- Sorting
- Term Frequency
- Inverse Document Frequency
- Student Data Base
- Max Temperature
- Different ways of joining data
- Word Co-Occurrence

MR algorithm s (Graph)

- PageRank
- Inverted Index

Higher Level Abstractions for MR (Pig)

- Introduction and Architecture
- Different Modes of executing Pig constructs Data Types
- Dynamic invokers Pig streaming Macros
- Pig Latin language Constructs (LOAD, STORE, DUMP, SPLI T, etc) User Defined Functions
- Use Cases

Higher Level Abstractions for MR (Hive)

- Introduction and Architecture
- Different Modes of executing Hive queries Metastore Implementations
- HiveQL (DDL & DML Operations) External v/s Managed Tables Views
- Partitions & Buckets User Defined Functions Transformations using Non Java Use Cases

Comparison of Pig and Hive

NoSQL Databases - 1 (Theoretical Concepts)

NoSQL Concepts

- Review of RDBMS
- Need for NoSQL
- Brewers CAP Theorem
- ACI D v/s BASE
- Schema on Read vs. Schema on Write
- Different levels of consistency
- Bloom filters

Different types of NoSQL databases
- Key Value
- Columnar
- Document
- Graph

Columnar Databases concepts NoSQL Databases - 2 (Practice) HBase Architecture
- Master and the Region Server
- Catalog tables (ROOT and META)
- Major and Minor compaction
- Configuration files
- HBase v/s Cassandra

Interfaces to HBase (for DDL and DML operations)
- Java API
- Client API
- Filters
- Scan Caching and Batching
- Com m and Line Interface
- REST API

Advance HBase Features
- HBase Data Modeling
- Bulk loading data in HBase
- HBase Coprocessors - EndPoints (similar to Stored Procedur es in RDBMS)
- HBase Coprocessors - Observers (similar to Triggers in RDBMS)

Spark
- Introduction to RDD
- Installation and Configuration of Spark
- Spark Architecture
- Different interfaces to Spark
- Sample Python programs in Spark

Setting up a Hadoop Cluster using Apache Hadoop
- Cloudera Hadoop cluster on the Amazon Cloud (Practice)
- Using EMR (Elastic Map Reduce)

Using EC2 (Elastic Compute Cloud)

SSH Configuration
- Stand alone m ode (Theory) Distributed m ode (Theory)
- Pseudo distributed
- Fully distributed

Hadoop Ecosystem and Use Cases

- Hadoop industry solutions
- Importing/exporting data across RDBMS and HDFS using Sqoop Getting real- time events into HDFS using Flume
- Creating workflows in Oozie Introduction to Graph processing Graph processing with Neo4J
- Processing data in real time using Storm
- Interactive Adhoc querying with Impala

Proof of concepts and use cases

- Click Stream Analysis using Pig and Hive Analyzing the Twitter data with Hive
- Further ideas for data analysis

CASE STUDY # 1 – "Twitter Analysis"

It is generally observed that about 80% of the data is unstructured while only 20% is actually structured data. RDBMS helps store/process this structured data, while Hadoop stores/processes both structured and unstructured data.

Twitter is fast becoming a true source of data to figure out what the customer is thinking something (sentimental analysis), to figure out what is currently trending, and so much more. In this case study, data is obtained from Twitter using various possible means and is analyzed.

CASE STUDY # 2 – "Click Stream Analysis"

Many e-Commerce sites have made quite an impact on the overall economy of many countries. All e-Commerce portals store data for user activities that happen on their site as Clickstream activity. This data is later analyzed to identify what the user has browsed and give recommendations to the user based on this or send out a personalized email to them. In this case study, we see how the Clickstream data and user data can be analyzed using Pig and Hive. The study derives data from RDBMS and the user behavior data (Clickstream) using Flume into HDFS, and is then analyzed using both Pig and Hive. The Clickstream analysis would also be automated using the workflow engine Oozie.

Course Duration: 60 hours

Prerequisite: Core Java/Linux and SQL

Spark Development Training Course Content

join@cognixia.com www.cognixia.com

Cognixia
A Collabera LEARNING SOLUTIONS COMPANY

Spark & Scala Development Training

Topics covered in the training

Prerequisite: SCALA for Spark Scala Basics
What is Scala?
- Why Scala for Spark?
- Intro to Scala REPL : Journey from Java to Scala
- Installing Scala IDE
- Basic Operations
- Defining Functions

Scala Essentials
- Control Structures in Scala
- Loops - ForEach, While, Do-While
- Collections – Array, ArrayBuffer, Map, Tuples, Lists
- If Statements
- Conditional Operators
- Enumerations

OOP's and FP
- Class and Object Basics
- Scala Constructors
- Nested Classes
- Visibility Rules
- Overriding Methods
- Functional Programming
- Higher Order Functions
- Traits
- Interfaces
- Layered Traits

Prerequisite: BigData and Hadoop Framework
- Introduction to BigData
- Challenges with Bigdata
- Batch Vs. Realtime processing

Overview- Hadoop Ecosystem
- HDFS

Spark & Scala Development Training
- Review of MapReduce
- Hive
- Sqoop
- Flume

APACHE SPARK
Introduction to Spark
- What is Spark?
- Spark Overview
- Setting up environment
- Using Spark Shell
- Spark Web UI

Spark Basics
- RDDs
- Spark Context
- Spark Ecosystem
- In-Memory data - Spark

Working with RDDs
- Creating, Loading and Saving RDD
- Transformations in RDD
- Actions in RDD
- Key-Value Pair RDD
- MapReduce and Pair RDD operations
- RDD Partitions

Writing and Deploying Spark Applications
- Spark Applications vs. Spark Shell
- Creating Spark Context
- Building a Spark Application
- Running a Spark Application
- Spark and Hadoop Integration-HDFS
- Handling Sequence Files

LEARN. ACHIEVE. STANDOUT

Spark RDD
- RDD Lineage
- RDD Persistence Overview
- Distributed Persistence

Spark SQL
- Overview of Hive
- Spark SQL Architecture
- SQLContext in Spark SQL
- Working with
- DataFrames
- Example for Spark SQL
- Integrating Hive and Spark SQL
- DataFrames,Datasets and RDD's
- Caching dataframes
- Knowing JSON and Parquet File Formats
- Loading of data
- Comparing Spark SQL,Impala and Hive-on-Spark

Spark Job Execution
- Jobs, Stages and Tasks
- Partition and shuffles
- Data Locality

Spark Streaming
- Spark Streaming Architecture
- First Spark Streaming programming
- Transformations in Spark Streaming

Spark Mllib
- What is Machine Learning?
- ML library for Spark
- ML Algorithms
- ML using Pipelines and DataFrames

GraphX
- Overview of GraphX
- Components of GraphX
- Hands-on - PageRank, TriangleCount
- Common Spark use-cases

Performance Tuning
- Shared Variables: Broadcast Variables
- Shared Variables: Accumulators
- Common Performance Issues
- Performance tuning tips

Course Deliverables
- Workshop style coaching
- Interactive approach
- Course material
- POC Implementation
- Hands on practice exercises for each topic
- Quiz at the end of each major topic
- Tips and techniques on Cloudera Certification Examination
- Linux concepts and basic commands

MongoDB
Course Content

join@cognixia.com www.cognixia.com

Cognixia
A Collabera LEARNING SOLUTIONS COMPANY

Introduction to NoSQL and MongoDB
RDBMS, types of relational databases, challenges of RDBMS, NoSQL database, its significance, how NoSQL suits Big Data needs, Introduction to MongoDB and its advantages, MongoDB installation, JSON features, data types and examples.

MongoDB Installation
Installing MongoDB, basic MongoDB commands and operations, MongoChef (MongoGUI) Installation, MongoDB Data types.

Importance of NoSQL
The need for NoSQL, types of NoSQL databases, OLTP, OLAP, limitations of RDBMS, ACID properties, CAP Theorem, Base property, learning about JSON/BSON, database collection & document, MongoDB uses, MongoDB Write Concern – Acknowledged, Replica Acknowledged, Unacknowledged, Journaled, Fsync.

CRUD Operations
Understanding CRUD and its functionality, CRUD concepts, MongoDB Query & Syntax, read and write queries and query optimization.

Data Modeling & Schema Design
Concepts of data modeling, difference between MongoDB and RDBMS modeling, Model tree structure, operational strategies, monitoring and backup.

Data Management & Administration
In this module you will learn MongoDB® Administration activities such as Health Check, Backup, Recovery, database sharding and profiling, Data Import/Export, Performance tuning etc.

Data Indexing and Aggregation
Concepts of data aggregation and types, data indexing concepts, properties and variations.

MongoDB Security
Understanding database security risks, MongoDB security concept and security approach, MongoDB integration with Java and Robomongo.

Working with Unstructured Data
Implementing techniques to work with variety of unstructured data like images, videos, log data, and others, understanding GridFS MongoDB file system for storing data.

MongoDB Project

Java is one of the most popular programming languages for working with MongoDB. This project tells you how to work with the MongoDB Java Driver, and using MongoDB as a Java Developer. Become proficient in creating a table for inserting Video using Java programming. Some of the tasks and steps involved are-

- Installation of Java
- Setting up MongoDB JDBC Driver
- Connecting to the database
- Understanding about collections and documents
- Reading and writing basics from the database
- Learning about the Java Virtual Machine libraries