



# Apache Spark and Scala Certification Training

---

# Table of Contents

1. About the Program
2. About Intellipaat
3. Key Features
4. Career Support
5. Why take up this course?
6. Who should take up this course?
7. Program Curriculum
8. Project Work
9. Certification
10. Intellipaat Success Stories
11. Contact Us



---

## About the Program

Intellipaat's Spark training lets you master real-time data processing using Spark Streaming, Spark SQL, Spark RDDs, and Spark Machine Learning libraries (Spark MLlib). You will learn Spark and Scala programming and will work on three real-life use cases in this Spark and Scala course.

---

## About Intellipaat

Intellipaat is one of the leading e-learning training providers with more than 600,000 learners across 55+ countries. We are on a mission to democratize education as we believe that everyone has the right to quality education.

Our courses are delivered by subject matter experts from top MNCs, and our world-class pedagogy enables learners to quickly learn difficult topics in no time. Our 24/7 technical support and career services will help them jump-start their careers in their dream companies.

# Key Features



**24 HRS INSTRUCTOR-LED  
TRAINING**



**22 HRS SELF-PACED TRAINING**



**60 HRS REAL-TIME  
PROJECT WORK**



**LIFETIME ACCESS**



**24/7 TECHNICAL SUPPORT**



**INDUSTRY-RECOGNIZED  
CERTIFICATION**



**JOB ASSISTANCE THROUGH  
80+ CORPORATE TIE-UPS**



**FLEXIBLE SCHEDULING**



# Career Support



## SESSIONS WITH INDUSTRY MENTORS

Attend sessions from top industry experts and get guidance on how to boost your career growth



## MOCK INTERVIEWS

Mock interviews to make you prepare for cracking interviews by top employers



## GUARANTEED INTERVIEWS & JOB SUPPORT

Get interviewed by our 400+ hiring partners



## RESUME PREPARATION

Get assistance in creating a world-class resume from our career services team



---

## Why take up this course?

- Apache Spark is an open-source computing framework that is up to 100 times faster than MapReduce
- Spark is an alternative form of data processing, which is unique in batch processing and streaming
- This is a comprehensive course for advanced implementation of Scala
- It helps you prepare yourself for Cloudera Hadoop Developer and Spark Professional Certification
- Get professional credibility to your resume so that you get hired faster with a high salary

---

## Who should take up this course?

- Software Engineers looking to upgrade in Big Data skills
- Data Engineers and ETL Developers
- Data Scientists and Analytics Professionals
- Graduates who are looking to make a career in Big Data

---

# Program Curriculum

## Spark and Scala Course Content

### Scala Course Content

#### 1. INTRODUCTION TO SCALA

- 1.1 Introducing Scala
- 1.2 Deployment of Scala for Big Data applications and Apache Spark analytics
- 1.3 Scala REPL, lazy values, and control structures in Scala
- 1.4 Directed Acyclic Graph (DAG)
- 1.5 First Spark application using SBT/Eclipse
- 1.6 Spark Web UI
- 1.7 Spark in the Hadoop ecosystem

#### 2. PATTERN MATCHING

- 2.1 The importance of Scala
- 2.2 The concept of REPL (Read Evaluate Print Loop)
- 2.3 Deep dive into Scala pattern matching
- 2.4 Type interface, higher-order function, currying, traits, application space, and Scala for data analysis

#### 3. EXECUTING THE SCALA CODE

- 3.1 Learning about Scala Interpreter
- 3.2 Static object timer in Scala and testing string equality in Scala
- 3.3 Implicit classes in Scala
- 3.4 The concept of currying in Scala
- 3.5 Various classes in Scala

#### 4. CLASSES CONCEPT IN SCALA

- 4.1 Learning about the classes concept
- 4.2 Understanding the constructor overloading
- 4.3 Various abstract classes
- 4.4 Hierarchy types in Scala

4.5 The concept of object equality

4.6 The val and var methods in Scala

## **5. CASE CLASSES & PATTERN MATCHING**

5.1 Understanding sealed traits, wild, constructor, tuple, variable pattern, and constant pattern

## **6. CONCEPTS OF TRAITS WITH EXAMPLES**

6.1 Understanding traits in Scala

6.2 Advantages of traits

6.3 Linearization of traits

6.4 The Java equivalent

6.5 Avoiding of boilerplate code

## **7. SCALA–JAVA INTEROPERABILITY**

7.1 Implementation of traits in Scala and Java

7.2 Handling of multiple traits extending

## **8. SCALA COLLECTIONS**

8.1 Introduction to Scala collections

8.2 Classification of collections

8.3 The difference between iterator and iterable in Scala

8.4 Example of list sequence in Scala

## **9. MUTABLE COLLECTIONS VS IMMUTABLE COLLECTIONS**

9.1 Two types of collections in Scala

9.2 Mutable and immutable collections

9.3 Understanding lists and arrays in Scala

9.4 The list buffer and the array buffer

9.6 Queue in Scala

9.7 Double-ended queue, deque, stacks, sets, maps, and tuples in Scala

## **10. USE CASE of BOBSROCKETS PACKAGE**

10.1 Introduction to Scala packages and imports

10.2 Selective imports

10.3 Scala test classes



- 10.4 Introduction to JUnit test class
- 10.5 JUnit interface via JUnit 3 suite for Scala test
- 10.6 Packaging of Scala applications in the directory structure
- 10.7 Examples of Spark Split and Spark Scala

## **Spark Course Content**

### **11. INTRODUCTION TO SPARK**

- 11.1 Introduction to Spark
- 11.2 How Spark overcomes the drawbacks of working on MapReduce
- 11.3 Understanding in-memory MapReduce
- 11.4 Interactive operations on MapReduce
- 11.5 Spark stack, fine vs coarse-grained update, Spark Hadoop YARN, HDFS Revision, and YARN Revision
- 11.6 The overview of Spark and how it is better than Hadoop
- 11.7 Deploying Spark without Hadoop
- 11.8 Spark history server and Cloudera distribution

### **12. SPARK BASICS**

- 12.1 Spark installation guide
- 12.2 Spark configuration
- 12.3 Memory management
- 12.4 Executor memory vs driver memory
- 12.5 Working with Spark Shell
- 12.6 The concept of resilient distributed datasets (RDD)
- 12.7 Learning to do functional programming in Spark
- 12.8 The architecture of Spark

### **13. WORKING WITH RDDS IN SPARK**

- 13.1 Spark RDDs
- 13.2 Creating RDDs
- 13.3 RDD partitioning
- 13.4 Operations and transformations in RDDs
- 13.5 Deep dive into Spark RDDs
- 13.6 RDD general operations

13.7 Read-only partitioned collection of records

13.8 Using the concept of RDDs for faster and efficient data processing

13.9 RDD action for the collect, count, collect, map, save-as-text-files, and paired RDD functions

## **14. AGGREGATING DATA WITH PAIRED RDDS**

14.1 Understanding the concept of key-value pair in RDDs

14.2 Learning how Spark makes MapReduce operations faster

14.3 Various operations of RDDs

14.4 MapReduce interactive operations

14.5 Fine and coarse-grained update

14.6 Spark stack

## **15. WRITING & DEPLOYING SPARK APPLICATIONS**

15.1 Comparing Spark applications with Spark Shell

15.2 Creating a Spark application using Scala or Java

15.3 Deploying a Spark application

15.4 Scala built application

15.5 Creation of the mutable list, set and set operations, lists, tuples, and concatenating lists

15.6 Creating an application using SBT

15.7 Deploying an application using Maven

15.8 The web user interface of a Spark application

15.9 A real-world example of Spark

15.10 Configuring Spark

## **16. PARALLEL PROCESSING**

16.1 Learning about Spark parallel processing

16.2 Deploying on a cluster

16.3 Introduction to Spark partitions

16.4 File-based partitioning of RDDs

16.5 Understanding HDFS and data locality

16.6 Mastering the technique of parallel operations

16.7 Comparing repartition and coalesce

16.8 RDD actions

## **17. SPARK RDD PERSISTENCE**

17.1 The execution flow in Spark

17.2 Understanding RDD persistence

17.3 Spark execution flow and Spark terminology

17.4 Distributed shared memory vs RDD

17.5 RDD limitations

17.6 Spark shell arguments

17.7 Distributed persistence

17.8 RDD lineage

17.9 Key-value pair for sorting implicit conversions such as CountByKey, ReduceByKey, SortByKey, and AggregateByKey

## **18. SPARK MLlib**

18.1 Introduction to Machine Learning

18.2 Types of Machine Learning

18.3 Introduction to MLlib

18.4 Various ML algorithms supported by MLlib

18.5 Linear regression, logistic regression, decision tree, random forest, and k-means clustering techniques

### **Hands-on Exercise:**

1. Building a recommendation engine

## **19. INTEGRATING APACHE FLUME & APACHE KAFKA**

19.1 Why Kafka, and what is Kafka?

19.2 Kafka architecture

19.3 Kafka workflow

19.4 Configuring Kafka cluster

19.5 Operations

19.6 Kafka monitoring tools

19.7 Integrating Apache Flume and Apache Kafka

### **Hands-on Exercise:**

1. Configuring a single-node single-broker cluster

2. Configuring a single-node multi-broker cluster
3. Producing and consuming messages
4. Integrating Apache Flume and Apache Kafka

## **20. SPARK STREAMING**

- 20.1 Introduction to Spark Streaming
- 20.2 Features of Spark Streaming
- 20.3 Spark Streaming workflow
- 20.4 Initializing StreamingContext, Discretized Streams (DStreams), input DStreams, and receivers
- 20.5 Transformations on DStreams, output operations on DStreams, windowed operators, and why they are useful
- 20.6 Important windowed operators and stateful operators

### **Hands-on Exercise:**

1. Twitter sentiment analysis
2. Streaming using the Netcat server
3. Kafka–Spark streaming
4. Spark–Flume streaming

## **21. IMPROVING SPARK PERFORMANCE**

- 21.1 Introduction to various variables in Spark such as shared variables and broadcast variables
- 21.2 Learning about accumulators
- 21.3 Common performance issues
- 21.4 Troubleshooting performance problems

## **22. SPARK SQL & DATAFRAMES**

- 22.1 Learning about Spark SQL
- 22.2 The context of SQL in Spark for providing structured data processing
- 22.3 JSON support in Spark SQL
- 22.4 Working with XML data
- 22.5 Parquet files
- 22.6 Creating Hive context
- 22.7 Writing a DataFrame to Hive
- 22.8 Reading JDBC files

22.9 Understanding the DataFrames in Spark

22.10 Creating DataFrames

22.11 Manual inferring of schema

22.12 Working with CSV files

22.13 Reading JDBC tables

22.14 DataFrame to JDBC

22.15 User-defined functions in Spark SQL

22.16 Shared variables and accumulators

22.17 Learning to query and transform data in DataFrames

22.18 How a DataFrame provides the benefits of both Spark RDDs and Spark SQL

22.19 Deploying Hive on Spark as the execution engine

## **23. SCHEDULING/PARTITIONING**

23.1 Learning about the scheduling and partitioning in Spark

23.2 Hash partition

23.3 Range partition

23.4 Scheduling within and around applications

23.5 Static partitioning, dynamic sharing, and fair scheduling

23.6 Map partition with index, Zip, and GroupByKey

23.7 Spark master high availability, standby masters with ZooKeeper, single-node recovery with the local file system, and high-order functions

---

# Project Work

## Spark and Scala Projects

### 1. Movie Recommendation

In this project, you will deploy Apache Spark for a movie recommendation system. Here, you will be working with Spark MLlib, collaborative filtering, clustering, regression, and dimensionality reduction. By the completion of this project, you will be proficient in working with streaming data, sampling, testing, and statistics.

### 2. Twitter API Integration for Tweet Analysis

Here, you will integrate Twitter API for analyzing tweets. You can use any of the scripting languages, such as PHP, Ruby, or Python, for requesting the Twitter API and get the results in the JSON format. You will have to perform aggregation, filtering, and parsing as per the requirement for the tweet analysis.

### 3. Data Exploration Using Spark SQL – Wikipedia Dataset

This project will allow you to work with Spark SQL and combine it with ETL applications, real-time analysis of data, performing batch analysis, deploying Machine Learning, creating visualizations, and the processing of graphs.



# Certification

After the completion of the course, you will get a certificate from Intellipaat.



## CERTIFICATE OF COMPLETION

This certificate is awarded to

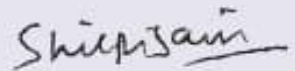
Your Name

Who has successfully completed

**Course Name**

Fulfilling all the requirements stipulated by Intellipaat to achieve professional excellence.

Issued Date: Month XX, XXXX



Mrs. Shilpi Jain  
Director,  
intellipaat Software Solutions Pvt. Ltd.

VERIFIED  
CERTIFICATE

Certificate ID #94658291

---

## Success Stories



**Kevin K Wada**

Thank you very much for your top-class service. A special mention should be made for your patience in listening to my queries and giving me a solution, which was exactly what I was looking for. I am giving you a 10 on 10!



**Sampson Basoah**

The Intellipaate team helped me in selecting the perfect course that suits my profile. The whole course was practically oriented, and the trainers are always ready to answer any question. I found this course to be impactful. Thank you.



**Suman Gajavelly**

I firmly believe that Intellipaate is the perfect place to embark on a great career in the technology space. Their Apache Spark and Scala training course was praiseworthy.



**Anthony Crenshaw**

I am glad that I took Intellipaate's Spark training. The trainers offered quality Spark training with real-world examples, and there was extensive interactivity throughout the training that made it the best, according to me.

---

## CONTACT US

### INTELLIPAAT SOFTWARE SOLUTIONS PVT. LTD.

#### Bangalore

AMR Tech Park 3, Ground Floor, Tower B,  
Hongasandra Village, Bommanahalli,  
Hosur Road, Bangalore – 560068

#### USA

1219 E. Hillsdale Blvd. Suite 205,  
Foster City, CA 94404

If you have any further queries or just want to have a conversation with us, then do call us.

**IND: +91-7022374614 | US: 1-800-216-8930**