**Scala Training**

Chinnasamy

[chinnasamyad@gmail.com](mailto:chinnasamyad@gmail.com)

**Audience**

The course coverage and pace would vary slightly, depending on the composition of the batch. If the training is for participants who are already familiar with some other object-oriented programming language, such as C++ or Java, the initial parts covering the basic language constructs as well as introduction to the OO concepts could be completed faster, and more time could be spent on some of the advanced aspects of the course. If the training is for a batch of participants who are new to any programming language, then even the basic language constructs would require more detailed explanation and practice work, and coverage of some of the later, advanced topics would be covered at the end.

**Day - 1**

**[1] Introduction to Scala**

[1.1] Why Scala?

[1.2] What is Scala?, Introducing Scala, Installing Scala, Journey - Java to Scala

[1.3] First Dive - Interactive Scala, Writing and Compiling Scala Programs

[1.4] Scala - REPL

[1.5] Scala Basics and Scala Basic Types

[1.6] Defining functions - Functions are first class citizens

[1.7] Imperative languages vs Functional Languages

[1.8] IDE for Scala, Scala Community.

[1.9] About the IntelliJ IDE. Setting up the IDE for the scala development.

**[2] Scala Essentials**

[2.1] Immutability in Scala.

[2.2] Semicolons and return statement.

[2.3] Method Declaration, Literals, Reserved Words, Operators, Precedence Rules, If

statements, While Loops, Do-While Loops, Conditional Operators.

[2.4] Enumerations.

[2.5] Factory Pattern using match keyword

**[3] Traits and OOPs in Scala**

[3.1] Traits - Traits as Mixins, Stackable Traits.

[3.2] Creating Traits Basic OOPS - Class and Object Basics.

[3.3] Class Constructors, Nested Classes, Visibility Rules.

**[4] Functional Programming in Scala**

[4.1] Topics - What is Functional Programming?, Functional Literals and Closures,

[4.2] Recursion, Tail Calls,

[4.3] Functional Data Structures,

[4.4] Implicit Function Parameters - Implicit values, Implicit Conversions and Implicit

classes.

[4.5] Call by Name, Call by Value.

**[5] Functional Programming**

[5.1] Map Transformation.

[5.2] Writing a functional literal (lambda expression) in a map transformation.

[5.3] map, flatMap, reduce, filter, head, take, drop, reduceLeft, fold, foldLeft, zip

transformations.

[5.4] Writing different types of functions.

**Day 2**

**[6] Variable Arguments**

[6.1] Discussion on the \_\* type.

[6.2] Usage of underscore in different places.

**[7] Collections**

[7.1] List

[7.2] Set

[7.3] Tuple

[7.4] Range

[7.5] Arrays

[7.6] Mutable

[7.7] Immutable

[7.8] Parallelized Collections

[7.9] Collection Transformations

**[8] Variants**

[8.1] Covariant

[8.2] Contravariant

[8.3] Invariant Generic Types

**[9] Currying Functions**

[9.1] Detailed study and usage of currying and partial applied functions.

**[10] XML Manipulating in Scala**

[10.1] Working with XML literals in code

[10.2] Embedding XPath like expressions

[10.3] Using Pattern Matching to process XML data

[10.4] Serializing and deserializing to and from XML

**[11] Build**

[11.1] Elucidation on maven and sbt

[11.2] Coding a maven pom file

[11.3] Coding a sbt

**[12] Leftovers**

[12.1] Bounded Types

[12.2] isInstanceOf and asInstanceOf

[12.3] Usage of annotations - concise code

[12.4] Sealed classes

[12.5] Option Class

[12.6] Building a jar using maven or sbt

**Notable Corporate Trainings**

|  |  |
| --- | --- |
| **Company Name** | **Trainings** |
| Wipro | Apache Spark, Scala, ELK |
| IBM (Hyderabad, Bangalore) | Apache Spark, Scala |
| HP | Scala |
| Scholastic | Apache Spark |
| HCL | Apache Spark, Hadoop, ELK Stack |
| ANZ | Spark, Scala |
| ITC Infotech | Spark, Scala, SBT, Kafka |
| Relevance Lab | Elasticsearch, Spark |
| Big Data Analytics Private Ltd | Hadoop, Spark |

Project: A live project of how each of the API’s are used in the industry.

Hands-on/Lecture Ratio:

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

Note to participants:

[\*] All content in this course will be a hands-on session.

[\*] All slides of the course will be given to candidates.

[\*] Source code of all examples tried out in the session will be provided.

Developer Environment:

[\*] OS can be either Windows or Linux

[\*] Either IntellijIDE or Eclipse (Scala Plugin installed)

[\*] Install scala on the developer machine

Scala 2.10, 2.11 and 2.12 should be installed.

[\*] Sbt also to be installed.

[\*] You can also install Scala-IDE for eclipse which you can download it

from <http://scala-ide.org/download/sdk.html>

[\*] The trainer has a MAC laptop, so infrastructure should be provided to

connect MAC laptop to the screen.