

# Industrial Training on Scala

## Chapter 1

### Getting familiar with IntelliJ IDEA

1. Introduction
2. Scala Environment Setup – Install Java Development Kit (JDK)
3. Scala Environment Setup – How to install the IntelliJ IDE
4. Scala Environment Setup – How to install Scala plugin in IntelliJ
5. Your first Scala Hello World application
6. Launch Scala application from main method
7. Running Your Scala application
8. Debugging Your Scala Application
9. Getting Started aWith Scala Project
10. Scala Project Classpath
11. Getting Started With build.sbt
12. SBT Dependencies And Maven
13. IntelliJ Settings and Preferences
14. IntelliJ Navigation Keyboard Shortcuts
15. IntelliJ Search Keyboard Shortcuts
16. IntelliJ Keyboard Shortcuts – Compile, Debug, Run

## Chapter 2

### Learning Scala: Fundamentals

1. Introduction
2. Variables And Types
3. String Interpolation
4. How To Escape Characters And Create Multi-line String
5. Type Inference
6. If Else Statement And Expression
7. For Comprehension
8. Range
9. While And Do While Loop
10. Pattern Matching
11. Tuples
12. Option
13. Class And Type Hierarchy
14. Enumerations

## Chapter 3

### Thinking in terms of functions

1. Introduction
2. Functions
3. Function With Parameters
4. Function With Option Parameters
5. Function With Option Return Type
6. Function With Implicit Parameter
7. Implicit Function
8. Typed Function
9. Polymorphic Function With Generic Return Type
10. Variable Argument Function
11. Functions As Symbols
12. Function Currying With Parameter Groups
13. Higher Order Function
14. Higher Order Function - Call By Name Function
15. Higher Order Function - With Callback Parameter
16. Function Using The Val Keyword Instead Of Def
17. Function Composition Using AndThen
18. Function Composition Using Compose
19. Tail Recursive Function - `@annotation.tailrec`
20. Tail Recursive Function - `scala.util.control.TailCalls._`
21. Trampoline Tail Recursive Function Using `scala.util.control.TailCalls._`
22. Partial Function Using the PartialFunction Trait
23. Nested Function

## Chapter 4

### Object Oriented using classes

1. Introduction
2. Classes And Objects
3. Companion Objects
4. Companion Objects' Apply Method As A Factory (Class Hierarchy Via Inheritance)
5. Values And Fields In Companion Object
6. Singleton Object
7. Case Class
8. Type Alias: Type Aliasing Versus Case Class
9. Implicit Class - Extension Methods
10. Package Objects
11. Extend Abstract Class
12. Case Class Inheritance
13. Typed Class
14. Covariance
15. Contra-Variance

## Chapter 5

## Traits and Dependency Injection

1. [Introduction](#)
2. [Create And Extend Trait](#)
3. [Trait With Type Parameters](#)
4. [Extend Multiple Traits](#)
5. [Dependency Injection Using Traits: Part 1](#)
6. [Dependency Injection Using Traits: Part 2 Avoid Cake Pattern](#)
7. [Traits, Companion Objects, Factory Pattern](#)

### Chapter 6

## Immutable Collection

1. [Introduction](#)
2. [List](#)
3. [ListSet](#)
4. [ListMap](#)
5. [Map](#)
6. [HashMap](#)
7. [TreeMap](#)
8. [Queue](#)
9. [Sequence](#)
10. [Set](#)
11. [HashSet](#)
12. [TreeSet](#)
13. [SortedSet](#)
14. [BitSet](#)
15. [Stack](#)
16. [Stream](#)
17. [Vector](#)

### Chapter 7

## Mutable Collection

1. [Introduction](#)
2. [Array](#)
3. [ArrayBuffer](#)
4. [ArrayStack](#)
5. [ListBuffer](#)
6. [Map](#)
7. [HashMap](#)

8. ListMap
9. LinkedHashMap
10. Queue
11. PriorityQueue
12. Set
13. HashSet
14. SortedSet
15. TreeSet
16. LinkedHashMap
17. BitSet

## Chapter 8

### Collection Functions

- Introduction

Function: aggregate	Function: collect
Function: diff	Function: drop
Function: dropWhile	Function: exists
Function: filter & filterNot	Function: find
Function: flatMap	Function: flatten
Function: fold	Function: foldLeft
Function: foldRight	Function: foreach
Function: groupBy	Function: head

Function: isEmpty	Function: intersect
Function: last	Function: map
Function: max	Function: maxBy
Function: min	Function: minBy
Function: mkString	Function: nonEmpty
Function: par	Function: partition
Function: reduce	Function: reduceLeft
Function: reduceRight	Function: reverse
Function: reverseIterator	Function: scan
Function: scanLeft	Function: scanRight
Function: size	Function: slice
Function: sortBy	Function: sorted
Function: sortWith	Function: tail

Function: take	Function: takeRight
Function: takeWhile	Function: transpose
Function: union	Function: unzip
Function: unzip3	Function: view
Function: withFilter	Function: zip
Function: zipWithIndex	

## Chapter 9 Futures

1. Introduction
2. Method with future as return type
3. Non blocking future result
4. Chain futures using flatMap()
5. Chain futures using for comprehension
6. Future Option with for comprehension
7. Future Option with map
8. Composing Futures
9. Future Sequence
10. Future Traverse
11. Future foldLeft
12. Future reduceLeft
13. Future firstCompletedOf
14. Future zip
15. Future zipWith
16. Future andThen
17. Future configure threadpool
18. Future recover
19. Future recoverWith
20. Future fallbackTo
21. Future promise

## Chapter 10

### Scala Test and Cheatsheet

#### ScalaTest:

1. [Introduction](#)
2. [Add ScalaTest as dependency to build.sbt](#)
3. [Create a test class using FlatSpec and Matchers](#)
4. [Equality Test](#)
5. [Length Test](#)
6. [Boolean Test](#)
7. [Collection Test](#)
8. [Type Test](#)
9. [Exception Test](#)
10. [Private Method Test](#)
11. [Future Method Test](#)

#### Collection:

1. [Convert Java collection](#)

#### General:

1. [Add line break](#)
2. [Convert multi-line string to single line](#)
3. [Check value of an Option](#)
4. [Read file content as a String](#)
5. [Create enum using sealed trait](#)
6. [Int division and return float](#)

#### Futures:

1. [Cannot find an implicit ExecutionContext](#)

## Chapter 11

### Learn SBT

#### Install SBT:

1. [Install SBT on Windows](#)
2. [Install SBT on Mac](#)

#### sbt run:

1. [Run a Scala main class](#)

#### sbt test:

1. [Run all tests](#)
2. [Run only one specific test](#)
3. [Run all integration tests](#)
4. [Run a specific integration test](#)
5. [Run one method in a test class](#)

#### **sbt debugging:**

1. [Analyze stacktrace in IntelliJ](#)

#### **sbt resources:**

1. [Show unmanaged classpaths](#)
2. [Show unmanaged resources](#)

### **Chapter 12 - Part 1**

## **Learn Apache Spark 2**

#### **Project Setup:**

1. [Using StackOverflow dataset](#)
2. [Add Apache Spark 2 SBT dependencies](#)
3. [Bootstrap a SparkSession](#)

#### **DataFrame SQL Query:**

1. [DataFrame Introduction](#)
2. [Create DataFrame from a CSV file](#)
3. [DataFrame schema](#)
4. [Select columns](#)
5. [Filter by column value](#)
6. [Count rows](#)
7. [SQL like](#)
8. [Filter chaining](#)
9. [SQL In](#)
10. [SQL Group By](#)
11. [SQL Group By with filter](#)
12. [SQL order by](#)
13. [Cast column data type](#)
14. [Filtered dataframe](#)
15. [Dataframe Join](#)
16. [Join and select columns](#)
17. [Join on explicit columns](#)
18. [Inner Join](#)
19. [Left Outer Join](#)
20. [Right Outer Join](#)
21. [Distinct](#)



## Chapter 12 - Part 2

### Learn Apache Spark 2

#### Spark SQL:

1. Spark SQL Introduction
2. Register temporary table
3. List all tables in Spark's catalog
4. List catalog tables using Spark SQL
5. Select columns
6. Filter by column value
7. Count number of rows
8. SQL like
9. SQL where with and clause
10. SQL In clause
11. SQL Group by
12. SQL Group by with having clause
13. SQL Order by
14. Typed columns, filter and create temp table
15. SQL Inner join
16. SQL Left outer join
17. SQL Right outer join
18. SQL Distinct
19. Register User Defined Function (UDF)

## Chapter 12 - Part 3

### Learn Apache Spark 2

#### DataFrame Statistics:

1. DataFrame Statistics Introduction
2. Create DataFrame from CSV
3. Average
4. Maximum
5. Minimum
6. Mean
7. Sum
8. Group by query with statistics
9. DataFrame Statistics using describe()
10. Correlation
11. Covariance
12. Frequent Items
13. Crosstab
14. Stratified sampling using sampleBy()

15. [Approximate Quantile](#)
16. [Bloom Filter](#)
17. [Count Min Sketch](#)
18. [Sampling With Replacement](#)

## **Chapter 12 - Part 4**

### Learn Apache Spark 2

#### **DataFrame Operations:**

1. [DataFrame Operations Introduction](#)
2. [Setup DataFrames](#)
3. [Convert DataFrame row to Scala Case class](#)
4. [Convert DataFrame row to Scala Case class using map\(\) method](#)
5. [Create DataFrame from Collection](#)
6. [DataFrame Union](#)
7. [DataFrame Intersection](#)
8. [Append column to DataFrame using withColumn\(\) method](#)

## **Chapter 13 - Part 2**

### Learn Akka

#### **Akka Actors & ActorSystems:**

#### **Akka Routers:**

1. [RoundRobinPool](#)
2. [ScatterGatherFirstCompletedPool](#)
3. [TailChoppingPool](#)
4. [BroadcastPool](#)

#### **Akka Dispatchers:**

1. [Akka Default Dispatcher](#)
2. [Akka Lookup Dispatcher](#)
3. [Fixed Thread Pool Dispatcher](#)
4. [Resizable Thread Pool Dispatcher](#)
5. [Pinned Thread Pool Dispatcher](#)

#### **Akka FSM:**

1. [Actor FSM become\(\)](#)
2. [Actor FSM unbecome\(\)](#)
3. [Actor FSM protocol](#)
4. [Actor LoggingFSM](#)

5. [Actor LoggingFSM Part Two](#)
6. [Actor LoggingFSM Part Three](#)
7. [Actor LoggingFSM Part Four](#)
8. [Actor LoggingFSM Part Five](#)