Scala Programming

Course Summary

Description

Scala may be the next wave of developer productivity. It is concise, object-oriented, functional, offers support for concurrency, but is compatible with Java and is JVM - based. For this reason Scala has been attracting the interest of many programming shops. However, these advantages are only possible through an effort of study. "Scala Programming" is designed to help this process through hands-on training.

Objectives

At the end of this course, students will be able to:

- Learn one of the hottest languages
- Be a productive programmer with Scala
- Learn to best software development practices with Scala

Topics

- Variables and Typing
- Collections
- Functions
- Classes
- Functional Programming
- Testing

Audience

This course is ideal for developers and architects.

Prerequisites

Prior to taking this course, students should be comfortable with Java programming language. Java knowledge may be substituted with C#, PHP, or similar.

Duration

Three days

ProTech protechtraining.com *"Charting the Course ... ... to Your Success!"*

Scala Programming

Course Outline

I. Brief History of Scala
- A. History of Java and JVM
- B. Pain points of Java
- C. Growth of JVM languages
- D. Motivation for Scala
- E. Why Scala was created

II. Basic Scala
- A. Object Oriented and Functional
- B. Declaring variables
- C. Typing
- D. Operators
- E. Conditionals
- F. Simple matchers
- G. Loops
- H. Scala Shell
- I. Labs

III. Collections
- A. Lists
- B. Arrays
- C. Tuples
- D. Sets
- E. Maps
- F. Labs

IV. Functions
- A. Functions
- B. Closures
- C. First-class functions
- D. Nested functions
- E. Recursion

V. Classes
- A. Classes and objects
- B. Declaring constructors
- C. Adding and overriding methods
- D. Traits
- E. Packages
- F. Access qualifiers
- G. Extending classes
- H. Annotations

VI. Exceptions
- A. Exception handling
- B. Declaring custom exceptions

VII. Java Interoperability
- A. Accessing java classes and methods

VIII. Functional Programming
- A. Closures
- B. Higher order functions
- C. Currying
- D. Partially applied functions

IX. Advanced Pattern Matching
- A. Classes in pattern matching
- B. Wild cards

X. Testing
- A. Testing frameworks
- B. Junit tests in Scala
- C. Behaviour driver test with ScalaTest
- D. Specs2

XI. Build Tools For Scala
- A. SBT

XII. Concurrency
- A. Actors
- B. Signals and Monitors
- C. Futures
- D. Parallel Computations
- E. Scala Future class and its use
- F. Better concurrency than threads and locks

XIII. Final
- A. Designing DSLs
- B. Play framework