Programming in Groovy
Duration: 3 days (9 hours each day with working lunch)

The power of Groovy is in metaprogramming. Sure, it's a language that's fluent, concise, integrates well with Java, and provides the functional style of programming. But, one of the main reasons to choose Groovy is to exploit its metaprogramming capabilities. If you're interested in learning about the compile time metaprogramming or the run-time dynamic capabilities or creating domain specific languages and fluent APIs with those capabilities, this course is specifically designed for you. While we get deeper into those capabilities of the language, along the way, we'll learn about various other features that make Groovy such a wonderful language to program in.

The course has a good balance of interactive lectures and hands-on exercises. The attendees are expected to pair-up and work on the lab exercises. The instructor will assist the attendees as they work on the labs. The objective of the course is for the attendees to gain an in depth practical knowledge of the concepts so they can put them to immediate use on their real projects.

The course content will be customized to meet your teams' specific needs. Please review this detailed outline and suggest changes (additions, deletions, modifications) as you feel fit.

Topics

Groovy—A quick start
* A dynamic, agile, OO language on the JVM
* Strengths of Groovy
* Programming with Groovy
* Scripting Groovy
* Compiling Groovy
* Creating JavaBeans
* Creating and using instances
* Implementing interfaces
* Operator Overloading
* Things to watch out with Groovy
* Exercises

Dynamic nature of Groovy
* Static typing vs. dynamic typing
* Benefits of static typing
* Limitations of static typing
* Power of dynamic typing
* Consequences of dynamic typing
* Groovy's optional typing
* Design by capability vs. design by contract
* When and how to specify types

* When to leave types out
* Polymorphism vs. multimethods
* Exercises

Functional style in Groovy

* Higher-order functions
* What's a closure?
* Creating closures
* Using closures
* Currying
* Dynamic closures
* Fluency through using closures
* Execute around method pattern using closures
* Strategy pattern using closures
* Exercises

Working with collections and strings

* Java collections through GDK fluency
* Convenience methods and iterators on lists
* Convenience methods and iterators on maps
* String expressions and evaluations
* Multiline strings
* String convenience methods
* Regular expressions
* Exercises

Runtime metaprogramming in Groovy

* Groovy Meta-object protocol
* Dynamically interacting with objects
* Interception vs. synthesis
* Ways to intercept methods
* Ways the synthesis methods
* Metaprogramming techniques
* Exercises

Compile time metaprogramming in Groovy

* Abstract syntax tree (AST) transformations
* Local vs. global transformation
* Phases of compilation
* AST transformation related annotations
* Creating AST transformations
* Injecting code using AST
* Exercises

Creating DSLs

* Domain specific languages
* Characteristics of DSLs
* Types of DSLs

* Designing DSLs
* Creating DSLs
* Realizing fluency
* Exercises

Integrating Groovy and Java

* Calling from Groovy into Java
* Implementing anonymous interfaces
* Calling from Java into Groovy
* Invoking Groovy dynamic methods from Java
* Invoking Groovy methods that take closures
* Exercises

Test driven development with Groovy

* Automated testing
* Facilities for automated testing
* Unit testing and functional testing
* Creating tests with Spock
* Setting expectations
* Data generators
* Mocking with Spock
* Creating functional tests with easyb
* Creating scenarios
* Integrating functional tests with underlying system
* Exercises

Creating web applications with Grails

* Grails and Groovy metaprogramming
* Architecture of a Grails application
* Views, Controllers, and domain classes
* Convention over configuration
* GORM
* Database mappings and configurations
* Customizing the mappings
* Exercises

Discussions

* Architectural concerns
* Pragmatics and application of Groovy and Grails
* Integrations related concerns and discussions
* Other aspects to consider

About the Instructor

Dr. Venkat Subramaniam is an award-winning author, founder of Agile Developer, Inc., creator of agilelearner.com, and an instructional professor at the University of Houston.

He has trained and mentored thousands of software developers in the US, Canada, Europe, and Asia, and is a regularly-invited speaker at several international conferences. Venkat helps his clients effectively apply and succeed with sustainable agile practices on their software projects.

Venkat is a (co)author of multiple technical books, including the 2007 Jolt Productivity award winning book Practices of an Agile Developer. You can find a list of his books at agiledeveloper.com. You may read more about Venkat and Agile Developer, Inc. at http://agiledeveloper.com.