Scala For Java Developers

Apache Spark 2.x for Java DevelopersModern Systems Programming with Scala NativeFunctional Programming, SimplifiedPragmatic ScalaScala Programming Projects97 Things Every Java Programmer Should KnowIntroduction to the Art of Programming Using ScalaModern Programming Made EasyProgramming with ScalaScala for Java DevelopersProgramming ScalaFunctional Programming in ScalaLearning AkkaThe Pragmatic ProgrammerLearning ScalaBeginning C++ Game ProgrammingScala for Java DevelopersClojure for Java DevelopersThe Type Astronaut's Guide to ShapelessFunctional Programming Patterns in Scala and ClojureProfessional ScalaFunctional ThinkingSpark in Action, Second EditionGroovy ProgrammingProgramming in ScalaBeginning ScalaFunctional Programming for Java DevelopersBecoming FunctionalLearn Scala for Java DevelopersThe Well-Grounded Java DeveloperScala for the ImpatientStream Processing with Apache FlinkHibernate Search by ExampleProgramming ScalaFunctional Programming in JavaBuilding Applications with ScalaLearning Concurrent Programming in ScalaA Beginner's Guide to Scala, Object Orientation and Functional ProgrammingPractical Scala DSLsScala in Depth

Apache Spark 2.x for Java Developers

What others in the trenches say about The Pragmatic Programmer “The cool thing about this book is that

it's great for keeping the programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there." —Kent Beck, author of Extreme Programming Explained: Embrace Change "I found this book to be a great mix of solid advice and wonderful analogies!" —Martin Fowler, author of Refactoring and UML Distilled "I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost." —Kevin Ruland, Management Science, MSG-Logistics "The wisdom and practical experience of the authors is obvious. The topics presented are relevant and useful. By far its greatest strength for me has been the outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the need for orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful information for journeymen programmers and expert mentors alike." —John Lakos, author of Large-Scale C++ Software Design "This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients." —Eric Vought, Software Engineer "Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book." —Pete McBreen, Independent Consultant "Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my

company time and money while helping me get my job done quicker! This should be a desktop reference for everyone who works with code for a living." —Jared Richardson, Senior Software Developer, iRenaissance, Inc. "I would like to see this issued to every new employee at my company." —Chris Cleeland, Senior Software Engineer, Object Computing, Inc. "If I'm putting together a project, it's the authors of this book that I want. . . . And failing that I'd settle for people who've read their book." —Ward Cunningham Straight from the programming trenches, The Pragmatic Programmer cuts through the increasing specialization and technicalities of modern software development to examine the core process--taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining anecdotes, thoughtful examples, and interesting analogies, The Pragmatic Programmer illustrates the best practices and major pitfalls of many different aspects of software development. Whether you're a new coder, an experienced programmer, or a

manager responsible for software projects, use these lessons daily, and you'll quickly see improvements in personal productivity, accuracy, and job satisfaction. You'll learn skills and develop habits and attitudes that form the foundation for long-term success in your career. You'll become a Pragmatic Programmer.

Modern Systems Programming with Scala Native

Get up to speed on Scala, the JVM language that offers all the benefits of a modern object model, functional programming, and an advanced type system. Packed with code examples, this comprehensive book shows you how to be productive with the language and ecosystem right away, and explains why Scala is ideal for today's highly scalable, data-centric applications that support concurrency and distribution. This second edition covers recent language features, with new chapters on pattern matching, comprehensions, and advanced functional programming. You'll also learn about Scala's command-line tools, third-party tools, libraries, and language-aware plugins for editors and IDEs. This book is ideal for beginning and advanced Scala developers alike. Program faster with Scala's succinct and flexible syntax Dive into basic and advanced functional programming (FP) techniques Build killer big-data apps, using Scala's functional combinators Use traits for mixin composition and pattern matching for data extraction Learn the sophisticated type system that combines FP and object-oriented programming concepts Explore Scala-specific

concurrency tools, including Akka Understand how to develop rich domain-specific languages Learn good design techniques for building scalable and robust Scala applications

Functional Programming, Simplified

This book is a must-have tutorial for software developers aiming to write concurrent programs in Scala, or broaden their existing knowledge of concurrency. This book is intended for Scala programmers that have no prior knowledge about concurrent programming, as well as those seeking to broaden their existing knowledge about concurrency. Basic knowledge of the Scala programming language will be helpful. Readers with a solid knowledge in another programming language, such as Java, should find this book easily accessible.

Pragmatic Scala

Software development today is embracing functional programming (FP), whether it's for writing concurrent programs or for managing Big Data. Where does that leave Java developers? This concise book offers a pragmatic, approachable introduction to FP for Java developers or anyone who uses an object-oriented language. Dean Wampler, Java expert and author of Programming Scala (O'Reilly), shows you how to apply FP principles such as immutability, avoidance of side-effects, and higher-order functions to your Java code. Each chapter provides exercises to help you practice what you've learned. Once you grasp the benefits of

functional programming, you'll discover that it improves all of the code you write. Learn basic FP principles and apply them to object-oriented programming Discover how FP is more concise and modular than OOP Get useful FP lessons for your Java type design—such as avoiding nulls Design data structures and algorithms using functional programming principles Write concurrent programs using the Actor model and software transactional memory Use functional libraries and frameworks for Java—and learn where to go next to deepen your functional programming skills

Scala Programming Projects

Discover unique features and powerful capabilities of Scala Programming as you build projects in a wide range of domains Key Features Develop a range of Scala projects from web applications to big data analysis Leverage full power of modern web programming using Play Framework Build real-time data pipelines in Scala with a Bitcoin transaction analysis app Book Description Scala is a type-safe JVM language that incorporates object-oriented and functional programming (OOP and FP) aspects. This book gets you started with essentials of software development by guiding you through various aspects of Scala programming, helping you bridge the gap between learning and implementing. You will learn about the unique features of Scala through diverse applications and experience simple yet powerful approaches for software development. Scala Programming Projects will help you build a number of

applications, beginning with simple projects, such as a financial independence calculator, and advancing to other projects, such as a shopping application and a Bitcoin transaction analyzer. You will be able to use various Scala features, such as its OOP and FP capabilities, and learn how to write concise, reactive, and concurrent applications in a type-safe manner. You will also learn how to use top-notch libraries such as Akka and Play and integrate Scala apps with Kafka, Spark, and Zeppelin, along with deploying applications on a cloud platform. By the end of the book, you will not only know the ins and outs of Scala, but you will also be able to apply it to solve a variety of real-world problems What you will learn Build, test, and package code using Scala Build Tool Decompose code into functions, classes, and packages for maintainability Implement the functional programming capabilities of Scala Develop a simple CRUD REST API using the Play framework Access a relational database using Slick Develop a dynamic web UI using Scala.js Source streaming data using Spark Streaming and write a Kafka producer Use Spark and Zeppelin to analyze data Who this book is for If you are an amateur programmer who wishes to learn how to use Scala, this book is for you. Knowledge of Java will be beneficial, but not necessary, to understand the concepts covered in this book.

97 Things Every Java Programmer Should Know

Scala is a concise, statically typed scripting language

that runs on the Java Virtual Machine. It is both a functional programming language and object-oriented language but its emphasis on functional programming sets it apart from Java. Learn Scala for Java Developers is for Java developers looking to transition to programming Scala. The book will help you translate the Java you already know into Scala and kick-start your productivity. What's Inside Tour Scala and learn the basic syntax, constructs and how to use the REPL Translate Java syntax that you already know into Scala Learn what Scala offers over and above Java, functional programming concepts and idioms Tips and advice useful when transitioning existing Java projects to Scala

Introduction to the Art of Programming Using Scala

Presents an introduction to the Scala programming language which is an abbreviated version of object-orientated programming combined with the power of concurrency capable of running on the Java Virtual Machine.

Modern Programming Made Easy

With its flexibility for programming both small and large projects, Scala is an ideal language for teaching beginning programming. Yet there are no textbooks on Scala currently available for the CS1/CS2 levels. Introduction to the Art of Programming Using Scala presents many concepts from CS1 and CS2 using a modern, JVM-based language that works well for both

programming in the small and programming in the large. The book progresses from true programming in the small to more significant projects later, leveraging the full benefits of object orientation. It first focuses on fundamental problem solving and programming in the small using the REPL and scripting environments. It covers basic logic and problem decomposition and explains how to use GUIs and graphics in programs. The text then illustrates the benefits of object-oriented design and presents a large collection of basic data structures showing different implementations of key ADTs along with more atypical data structures. It also introduces multithreading and networking to provide further motivating examples. By using Scala as the language for both CS1 and CS2 topics, this textbook gives students an easy entry into programming small projects as well as a firm foundation for taking on larger-scale projects. Many student and instructor resources are available at www.programmingusingscala.net

Programming with Scala

Summary The Spark distributed data processing platform provides an easy-to-implement tool for ingesting, streaming, and processing data from any source. In Spark in Action, Second Edition, you'll learn to take advantage of Spark's core features and incredible processing speed, with applications including real-time computation, delayed evaluation, and machine learning. Spark skills are a hot commodity in enterprises worldwide, and with Spark's

powerful and flexible Java APIs, you can reap all the benefits without first learning Scala or Hadoop. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Analyzing enterprise data starts by reading, filtering, and merging files and streams from many sources. The Spark data processing engine handles this varied volume like a champ, delivering speeds 100 times faster than Hadoop systems. Thanks to SQL support, an intuitive interface, and a straightforward multilanguage API, you can use Spark without learning a complex new ecosystem. About the book Spark in Action, Second Edition, teaches you to create end-to-end analytics applications. In this entirely new book, you'll learn from interesting Java-based examples, including a complete data pipeline for processing NASA satellite data. And you'll discover Java, Python, and Scala code samples hosted on GitHub that you can explore and adapt, plus appendixes that give you a cheat sheet for installing tools and understanding Spark-specific terms. What's inside Writing Spark applications in Java Spark application architecture Ingestion through files, databases, streaming, and Elasticsearch Querying distributed datasets with Spark SQL About the reader This book does not assume previous experience with Spark, Scala, or Hadoop. About the author Jean-Georges Perrin is an experienced data and software architect. He is France's first IBM Champion and has been honored for 12 consecutive years. Table of Contents PART 1 - THE THEORY CRIPPLED BY AWESOME EXAMPLES 1 So, what is Spark, anyway? 2 Architecture and flow 3 The majestic role of the

dataframe 4 Fundamentally lazy 5 Building a simple app for deployment 6 Deploying your simple app PART 2 - INGESTION 7 Ingestion from files 8 Ingestion from databases 9 Advanced ingestion: finding data sources and building your own 10 Ingestion through structured streaming PART 3 - TRANSFORMING YOUR DATA 11 Working with SQL 12 Transforming your data 13 Transforming entire documents 14 Extending transformations with user-defined functions 15 Aggregating your data PART 4 - GOING FURTHER 16 Cache and checkpoint: Enhancing Spark's performances 17 Exporting data and building full data pipelines 18 Exploring deployment

Scala for Java Developers

If your application source code is overly verbose, it can be a nightmare to maintain. Write concise and expressive, type-safe code in an environment that lets you build for the JVM, browser, and more. Key Features Expert guidance that shows you to efficiently use both object-oriented and functional programming techniques Understand functional programming libraries, such as Cats and Scalaz, and use them to augment your Scala development Perfectly balances theory and hands-on exercises, assessments, and activities Book Description This book teaches you how to build and contribute to Scala programs, recognizing common patterns and techniques used with the language. You'll learn how to write concise, functional code with Scala. After an introduction to core concepts, syntax, and writing example applications with scalac, you'll learn about the Scala Collections

API and how the language handles type safety via static types out-of-the-box. You'll then learn about advanced functional programming patterns, and how you can write your own Domain Specific Languages (DSLs). By the end of the book, you'll be equipped with the skills you need to successfully build smart, efficient applications in Scala that can be compiled to the JVM. What you will learn Understand the key language syntax and core concepts for application development Master the type system to create scalable type-safe applications while cutting down your time spent debugging Understand how you can work with advanced data structures via built-in features such as the Collections library Use classes, objects, and traits to transform a trivial chatbot program into a useful assistant Understand what are pure functions, immutability, and higher-order functions Recognize and implement popular functional programming design patterns Who this book is for This is an ideal book for developers who are looking to learn Scala, and is particularly well suited for Java developers looking to migrate across to Scala for application development on the JVM.

Programming Scala

Why learn Scala? You don't need to be a data scientist or distributed computing expert to appreciate this object-oriented functional programming language. This practical book provides a comprehensive yet approachable introduction to the language, complete with syntax diagrams, examples, and exercises. You'll start with Scala's core types and

syntax before diving into higher-order functions and immutable data structures. Author Jason Swartz demonstrates why Scala's concise and expressive syntax make it an ideal language for Ruby or Python developers who want to improve their craft, while its type safety and performance ensures that it's stable and fast enough for any application. Learn about the core data types, literals, values, and variables Discover how to think and write in expressions, the foundation for Scala's syntax Write higher-order functions that accept or return other functions Become familiar with immutable data structures and easily transform them with type-safe and declarative operations Create custom infix operators to simplify existing operations or even to start your own domain-specific language Build classes that compose one or more traits for full reusability, or create new functionality by mixing them in at instantiation

Functional Programming in Scala

A step-by-step guide for getting started with Hibernate Search, employing a practical example application that will make readers quickly learn and use the excellent search capabilities that the Hibernate Search tool offers.This book is for any Java developer who wants to quickly and easily add feature-rich search capabilities to his/her custom applications. It is assumed that readers will have experience in Java development, and understand some basic relational database concepts. However, while core Hibernate experience is very helpful, it is not necessarily required.

Learning Akka

Groovy Programming is an introduction to the Java-based scripting language Groovy. Groovy has much in common with popular scripting languages such as Perl, Python, and Ruby, but is written in a Java-like syntax. And, unlike these other languages, Groovy is sanctioned by the Java community for use on the Java platform. Since it is based on Java, applications written in Groovy can make full use of the Java Application Programmer Interfaces (APIs). This means Groovy can integrate seamlessly with applications written in Java, while avoiding the complexities of the full Java language. This bare-bones structure also means Groovy can be used as an introduction to Java and to programming in general. Its simpler constructions and modern origins make it ideal as a first language and for introducing principles such as object-oriented programming. This book introduces all the major aspects of Groovy development and emphasizes Groovy's potential as a learning tool. Case studies and exercises are included, along with numerous programming examples. The book begins assuming only a general familiarity with Java programming, and progresses to discuss advanced topics such as GUI builders, Groovlets, Unit Testing, and Groovy SQL. The first comprehensive book on Groovy programming that shows how writing applications and scripts for the Java platform is fast and easy Written by leading software engineers and acclaimed computing instructors Offers numerous programming examples, code samples, detailed case studies, exercises for self-study, and a companion

website with a Windows-based Groovy editor

The Pragmatic Programmer

If you have an imperative (and probably object-oriented) programming background, this hands-on book will guide you through the alien world of functional programming. Author Joshua Backfield begins slowly by showing you how to apply the most useful implementation concepts before taking you further into functional-style concepts and practices. In each chapter, you'll learn a functional concept and then use it to refactor the fictional XXY company's imperative-style legacy code, writing and testing the functional code yourself. As you progress through the book, you'll migrate from Java 7 to Groovy and finally to Scala as the need for better functional language support gradually increases. Learn why today's finely tuned applications work better with functional code Transform imperative-style patterns into functional code, following basic steps Get up to speed with Groovy and Scala through examples Understand how first-class functions are passed and returned from other functions Convert existing methods into pure functions, and loops into recursive methods Change mutable variables into immutable variables Get hands-on experience with statements and nonstrict evaluations Use functional programming alongside object-oriented design

Learning Scala

Build fault tolerant concurrent and distributed

applications with Akka About This Book Build networked applications that self-heal Scale out your applications to handle more traffic faster An easy-to-follow guide with a number of examples to ensure you get the best start with Akka Who This Book Is For This book is intended for beginner to intermediate Java or Scala developers who want to build applications to serve the high-scale user demands in computing today. If you need your applications to handle the ever-growing user bases and datasets with high performance demands, then this book is for you. Learning Akka will let you do more for your users with less code and less complexity, by building and scaling your networked applications with ease. What You Will Learn Use Akka to overcome the challenges of concurrent programming Resolve the issues faced in distributed computing with the help of Akka Scale applications to serve a high number of concurrent users Make your system fault-tolerant with self-healing applications Provide a timely response to users with easy concurrency Reduce hardware costs by building more efficient multi-user applications Maximise network efficiency by scaling it In Detail Software today has to work with more data, more users, more cores, and more servers than ever. Akka is a distributed computing toolkit that enables developers to build correct concurrent and distributed applications using Java and Scala with ease, applications that scale across servers and respond to failure by self-healing. As well as simplifying development, Akka enables multiple concurrency development patterns with particular support and architecture derived from Erlang's concept of actors (lightweight concurrent entities). Akka is written in

Scala, which has become the programming language of choice for development on the Akka platform. Learning Akka aims to be a comprehensive walkthrough of Akka. This book will take you on a journey through all the concepts of Akka that you need in order to get started with concurrent and distributed applications and even build your own. Beginning with the concept of Actors, the book will take you through concurrency in Akka. Moving on to networked applications, this book will explain the common pitfalls in these difficult problem areas while teaching you how to use Akka to overcome these problems with ease. The book is an easy to follow example-based guide that will strengthen your basic knowledge of Akka and aid you in applying the same to real-world scenarios. Style and approach An easy-to-follow, example-based guide that will take you through building several networked-applications that work together while you are learning concurrent and distributed computing concepts. Each topic is explained while showing you how to design with Akka and how it is used to overcome common problems in applications. By showing Akka in context to the problems, it will help you understand what the common problems are in distributed applications and how to overcome them.

Beginning C++ Game Programming

Scala for Java Developers

Presents an introduction to the Scala programming

language which is an abbreviated version of object-orientated programming combined with the power of concurrency capable of running on the Java Virtual Machine.

Clojure for Java Developers

Presents an introduction to the new programming language for the Java Platform.

The Type Astronaut's Guide to Shapeless

Helps programmers learn functional programming and apply it to the everyday business of coding. Original.

Functional Programming Patterns in Scala and Clojure

If you want to push your Java skills to the next level, this book provides expert advice from Java leaders and practitioners. You'll be encouraged to look at problems in new ways, take broader responsibility for your work, stretch yourself by learning new techniques, and become as good at the entire craft of development as you possibly can. Edited by Kevlin Henney and Trisha Gee, 97 Things Every Java Programmer Should Know reflects lifetimes of experience writing Java software and living with the process of software development. Great programmers share their collected wisdom to help you rethink Java practices, whether working with legacy code or incorporating changes since Java 8. A few of the 97

things you should know: "Behavior Is Easy, State Is Hard"—Edson Yanaga “Learn Java Idioms and Cache in Your Brain”—Jeanne Boyarsky “Java Programming from a JVM Performance Perspective”—Monica Beckwith "Garbage Collection Is Your Friend"—Holly K Cummins “Java's Unspeakable Types”—Ben Evans "The Rebirth of Java"—Sander Mak “Do You Know What Time It Is?”—Christin Gorman

Professional Scala

Learn C++ from scratch and get started building your very own games About This Book This book offers a fun way to learn modern C++ programming while building exciting 2D games This beginner-friendly guide offers a fast-paced but engaging approach to game development Dive headfirst into building a wide variety of desktop games that gradually increase in complexity It is packed with many suggestions to expand your finished games that will make you think critically, technically, and creatively Who This Book Is For This book is perfect for you if any of the following describes you: You have no C++ programming knowledge whatsoever or need a beginner level refresher course, if you want to learn to build games or just use games as an engaging way to learn C++, if you have aspirations to publish a game one day, perhaps on Steam, or if you just want to have loads of fun and impress friends with your creations. What You Will Learn Get to know C++ from scratch while simultaneously learning game building Learn the basics of C++, such as variables, loops, and functions to animate game objects, respond to collisions, keep

score, play sound effects, and build your first playable game. Use more advanced C++ topics such as classes, inheritance, and references to spawn and control thousands of enemies, shoot with a rapid fire machine gun, and realize random scrolling game-worlds Stretch your C++ knowledge beyond the beginner level and use concepts such as pointers, references, and the Standard Template Library to add features like split-screen coop, immersive directional sound, and custom levels loaded from level-design files Get ready to go and build your own unique games! In Detail This book is all about offering you a fun introduction to the world of game programming, C++, and the OpenGL-powered SFML using three fun, fully-playable games. These games are an addictive frantic two-button tapper, a multi-level zombie survival shooter, and a split-screen multiplayer puzzle-platformer. We will start with the very basics of programming, such as variables, loops, and conditions and you will become more skillful with each game as you move through the key C++ topics, such as OOP (Object-Orientated Programming), C++ pointers, and an introduction to the Standard Template Library. While building these games, you will also learn exciting game programming concepts like particle effects, directional sound (spatialization), OpenGL programmable Shaders, spawning thousands of objects, and more. Style and approach This book offers a fun, example-driven approach to learning game development and C++. In addition to explaining game development techniques in an engaging style, the games are built in a way that introduces the key C++ topics in a practical and not theory-based way, with multiple runnable/playable

stages in each chapter.

Functional Thinking

Master the fundamentals of Scala and understand its emphasis on functional programming that sets it apart from Java. This book will help you translate what you already know in Java to Scala to start your functional programming journey. Learn Scala is split into four parts: a tour of Scala, a comparison between Java and Scala, Scala-specific features and functional programming idioms, and finally a discussion about adopting Scala in existing Java teams and legacy projects. After reading and using this tutorial, you'll come away with the skills in Scala to kick-start your productivity with this growing popular language. What You'll Learn Tour Scala and learn the basic syntax, constructs, and how to use the REPL Translate Java syntax that you already know into Scala Learn what Scala offers over and above Java Become familiar with functional programming concepts and idioms Gain tips and advice useful when transitioning existing Java projects to Scala Who This Book Is For Java developers looking to transition to Scala. No prior experience necessary in Scala.

Spark in Action, Second Edition

Summary The Well-Grounded Java Developer offers a fresh and practical look at new Java 7 features, new JVM languages, and the array of supporting technologies you need for the next generation of Java-based software. About the Book The Well-Grounded

Java Developer starts with thorough coverage of Java 7 features like try-with-resources and NIO.2. You'll then explore a cross-section of emerging JVM-based languages, including Groovy, Scala, and Clojure. You will find clear examples that are practical and that help you dig into dozens of valuable development techniques showcasing modern approaches to the dev process, concurrency, performance, and much more. Written for readers familiar with Java. No experience with Java 7 or new JVM languages required. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book. What's Inside New Java 7 features Tutorials on Groovy, Scala, and Clojure Discovering multicore processing and concurrency Functional programming with new JVM languages Modern approaches to testing, build, and CI Table of Contents PART 1 DEVELOPING WITH JAVA 7 Introducing Java 7 New I/O PART 2 VITAL TECHNIQUES Dependency Injection Modern concurrency Class files and bytecode Understanding performance tuning PART 3 POLYGLOT PROGRAMMING ON THE JVM Alternative JVM languages Groovy: Java's dynamic friend Scala: powerful and concise Clojure: safer programming PART 4 CRAFTING THE POLYGLOT PROJECT Test-driven development Build and continuous integration Rapid web development Staying well-grounded

Groovy Programming

This reader-friendly textbook presents a concise and easy to follow introduction to Scala. Scala is an ideal

first programming language, which permits programming in multiple paradigms, and enables developers to be more productive with modern computing infrastructures such as distributed environments. Topics and features: provides review questions and problem-solving exercises (with solutions) in each chapter, inspired by real-world applications; addresses each topic in a self-contained manner, highlighting how Scala can be evolved and grown according to the developer's needs; presents examples from a broad range of different application domains, including consumer electronics, online payment, retail, vehicle manufacturing, and healthcare; encourages an innovation-oriented mind-set, and the development of practical, saleable skills; draws from the author's extensive experience in industrial software development, academic research, and university teaching. This accessible and hands-on guide will embolden professional software engineers to make the switch to Scala. Instructors teaching introductory programming courses will also find this textbook popular among their students.

Programming in Scala

Provides a guide to using Scala and Clojure to solve in-depth programming problems.

Beginning Scala

If you're familiar with functional programming basics and want to gain a much deeper understanding, this in-depth guide takes you beyond syntax and

demonstrates how you need to think in a new way. Software architect Neal Ford shows intermediate to advanced developers how functional coding allows you to step back a level of abstraction so you can see your programming problem with greater clarity. Each chapter shows you various examples of functional thinking, using numerous code examples from Java 8 and other JVM languages that include functional capabilities. This book may bend your mind, but you'll come away with a much better grasp of functional programming concepts. Understand why many imperative languages are adding functional capabilities Compare functional and imperative solutions to common problems Examine ways to cede control of routine chores to the runtime Learn how memoization and laziness eliminate hand-crafted solutions Explore functional approaches to design patterns and code reuse View real-world examples of functional thinking with Java 8, and in functional architectures and web frameworks Learn the pros and cons of living in a paradigmatically richer world If you're new to functional programming, check out Josh Backfield's book Becoming Functional.

Functional Programming for Java Developers

If you've had trouble trying to learn Functional Programming (FP), you're not alone. In this book, Alvin Alexander -- author of the Scala Cookbook and former teacher of Java and Object-Oriented Programming (OOP) classes -- writes about his own problems in trying to understand FP, and how he

finally conquered it. What he originally learned is that experienced FP developers are driven by two goals: to use only immutable values, and write only pure functions. What he later learned is that they have these goals as the result of another larger goal: they want all of their code to look and work just like algebra. While that sounds simple, it turns out that these goals require them to use many advanced Scala features -- which they often use all at the same time. As a result, their code can look completely foreign to novice FP developers. As Mr. Alexander writes, "When you first see their code it's easy to ask, 'Why would anyone write code like this?'" Mr. Alexander answers that "Why?" question by explaining the benefits of writing pure functional code. Once you understand those benefits -- your motivation for learning FP -- he shares five rules for programming in the book: All fields must be immutable ('val' fields). All functions must be pure functions. Null values are not allowed. Whenever you use an 'if' you must also use an 'else'. You won't create OOP classes that encapsulate data and behavior; instead you'll design data structures using Scala 'case' classes, and write pure functions that operate on those data structures. In the book you'll see how those five, simple rules naturally lead you to write pure, functional code that reads like algebra. He also shares one more Golden Rule for learning: Always ask "Why"? Lessons in the book include: How and why to write only pure functions Why pure function signatures are much more important than OOP method signatures Why recursion is a natural tool for functional programming, and how to write recursive algorithms Because the Scala 'for' expression is so important to FP, dozens of pages

explain the details of how it works In the end you'll see that monads aren't that difficult because they're a natural extension of the Five Rules The book finishes with lessons on FP data modeling, and two main approaches for organizing your pure functions As Mr. Alexander writes, "In this book I take the time to explain all of the concepts that are used to write FP code in Scala. As I learned from my own experience, once you understand the Five Rules and the small concepts, you can understand Scala/FP." Please note that because of the limits on how large a printed book can be, the paperback version does not include all of the chapters that are in the Kindle eBook. The following lessons are not in the paperback version: Grandma's Cookies (a story about pure functions) The ScalaCheck lessons The Type Classes lessons The appendices Because those lessons didn' fit in the print version, they have been made freely available online. (Alvin Alexander (alvinalexander.com) wrote the popular Scala Cookbook for O'Reilly, and also self-published two other books, How I Sold My Business: A Personal Diary, and A Survival Guide for New Consultants.)

Becoming Functional

Transition smoothly from Java to the most widely used functional JVM-based language – Clojure About This Book Write apps for the multithreaded world with Clojure's flavor of functional programming Discover Clojure's features and advantages and use them in your existing projects The book is designed so that you'll be able put to use your existing skills and

software knowledge to become a more effective Clojure developer Who This Book Is For This book is intended for Java developers, who are looking for a way to expand their skills and understand new paradigms of programming. Whether you know a little bit about functional languages, or you are just getting started, this book will get you up and running with how to use your existing skills in Clojure and functional programming. What You Will Learn Understand the tools for the Clojure world and how they relate to Java tools and standards (like Maven) Learn about immutable data structures, and what makes them feasible for everyday programming Write simple multi-core programs using Clojure's core concepts, like atoms, agents and refs Understand that in Clojure, code is data, and how to take advantage of that fact by generating and manipulating code with macros Learn how Clojure interacts with Java, how the class loaders work and how to use Clojure from Java or the other way around Discover a new, more flexible meaning of polymorphism and understand that OOP is not the only way to get it In Detail We have reached a point where machines are not getting much faster, software projects need to be delivered quickly, and high quality in software is more demanding as ever. We need to explore new ways of writing software that helps achieve those goals. Clojure offers a new possibility of writing high quality, multi-core software faster than ever, without having to leave your current platform. Clojure for Java developers aims at unleashing the true potential of the Clojure language to use it in your projects. The book begins with the installation and setup of the Clojure environment before moving on to explore the language in-depth.

Get acquainted with its various features such as functional programming, concurrency, etc. with the help of example projects. Additionally, you will also, learn how the tooling works, and how it interacts with the Java environment. By the end of this book, you will have a firm grip on Clojure and its features, and use them effectively to write more robust programs. Style and approach An easy to follow, step-by-step, guide on how to start writing Clojure programs making use of all of its varied features and advantages. As this is a new language, certain new concepts are supported with theoretical section followed by simple projects to help you gain a better understanding and practice of how Clojure works.

Learn Scala for Java Developers

Scala is now an established programming language developed by Martin Oderskey and his team at the EPFL. The name Scala is derived from Sca(lable) La(nguage). Scala is a multi-paradigm language, incorporating object oriented approaches with functional programming. Although some familiarity with standard computing concepts is assumed (such as the idea of compiling a program and executing this compiled from etc.) and with basic procedural language concepts (such as variables and allocation of values to these variables) the early chapters of the book do not assume any familiarity with object orientation nor with functional programming These chapters also step through other concepts with which the reader may not be familiar (such as list processing). From this background, the book provides

a practical introduction to both object and functional approaches using Scala. These concepts are introduced through practical experience taking the reader beyond the level of the language syntax to the philosophy and practice of object oriented development and functional programming. Students and those actively involved in the software industry will find this comprehensive introduction to Scala invaluable.

The Well-Grounded Java Developer

The open source Scala language is a Java-based dynamic scripting, functional programming language. Moreover, this highly scalable scripting language lends itself well to building cloud-based/deliverable Software as a Service (SaaS) online applications. Written by Lift Scala web framework founder and lead Dave Pollak, Beginning Scala takes a down-to-earth approach to teaching Scala that leads you through simple examples that can be combined to build complex, scalable systems and applications. This book introduces you to the Scala programming language and then guides you through Scala constructs and libraries that allow small and large teams to assemble small components into high-performance, scalable systems. You will learn why Scala is becoming the language of choice for Web 2.0 companies such as Twitter as well as enterprises such as Seimens and SAP.

Scala for the Impatient

Unleash the data processing and analytics capability of Apache Spark with the language of choice: Java About This Book Perform big data processing with Spark—without having to learn Scala! Use the Spark Java API to implement efficient enterprise-grade applications for data processing and analytics Go beyond mainstream data processing by adding querying capability, Machine Learning, and graph processing using Spark Who This Book Is For If you are a Java developer interested in learning to use the popular Apache Spark framework, this book is the resource you need to get started. Apache Spark developers who are looking to build enterprise-grade applications in Java will also find this book very useful. What You Will Learn Process data using different file formats such as XML, JSON, CSV, and plain and delimited text, using the Spark core Library. Perform analytics on data from various data sources such as Kafka, and Flume using Spark Streaming Library Learn SQL schema creation and the analysis of structured data using various SQL functions including Windowing functions in the Spark SQL Library Explore Spark Mlib APIs while implementing Machine Learning techniques to solve real-world problems Get to know Spark GraphX so you understand various graph-based analytics that can be performed with Spark In Detail Apache Spark is the buzzword in the big data industry right now, especially with the increasing need for real-time streaming and data processing. While Spark is built on Scala, the Spark Java API exposes all the Spark features available in the Scala version for Java developers. This book will show you how you can implement various functionalities of the Apache Spark framework in Java, without stepping out of your

comfort zone. The book starts with an introduction to the Apache Spark 2.x ecosystem, followed by explaining how to install and configure Spark, and refreshes the Java concepts that will be useful to you when consuming Apache Spark's APIs. You will explore RDD and its associated common Action and Transformation Java APIs, set up a production-like clustered environment, and work with Spark SQL. Moving on, you will perform near-real-time processing with Spark streaming, Machine Learning analytics with Spark MLlib, and graph processing with GraphX, all using various Java packages. By the end of the book, you will have a solid foundation in implementing components in the Spark framework in Java to build fast, real-time applications. Style and approach This practical guide teaches readers the fundamentals of the Apache Spark framework and how to implement components using the Java language. It is a unique blend of theory and practical examples, and is written in a way that will gradually build your knowledge of Apache Spark.

Stream Processing with Apache Flink

This step-by-step guide is full of easy-to-follow code taken from real-world examples explaining the migration and integration of Scala in a Java project. If you are a Java developer or a Java architect, working in Java EE-based solutions and want to start using Scala in your daily programming, this book is ideal for you. This book will get you up and running quickly by adopting a pragmatic approach with real-world code samples. No prior knowledge of Scala is required.

Hibernate Search by Example

Get up and running fast with the basics of programming using Java as an example language. This short book gets you thinking like a programmer in an easy and entertaining way. Modern Programming Made Easy teaches you basic coding principles, including working with lists, sets, arrays, and maps; coding in the object-oriented style; and writing a web application. This book is largely language agnostic, but mainly covers the latest appropriate and relevant release of Java, with some updated references to Groovy, Scala, and JavaScript to give you a broad range of examples to consider. You will get a taste of what modern programming has to offer and set yourself up for further study and growth in your chosen language. What You'll Learn Write code using the functional programming style Build your code using the latest releases of Java, Groovy, and more Test your code Read and write from files Design user interfaces Deploy your app in the cloud Who This Book Is For Anyone who wants to learn how to code. Whether you're a student, a teacher, looking for a career change, or just a hobbyist, this book is made for you.

Programming Scala

Intermediate level, for programmers fairly familiar with Java, but new to the functional style of programming and lambda expressions. Get ready to program in a whole new way. Functional Programming in Java will help you quickly get on top of the new,

essential Java 8 language features and the functional style that will change and improve your code. This short, targeted book will help you make the paradigm shift from the old imperative way to a less error-prone, more elegant, and concise coding style that's also a breeze to parallelize. You'll explore the syntax and semantics of lambda expressions, method and constructor references, and functional interfaces. You'll design and write applications better using the new standards in Java 8 and the JDK. Lambda expressions are lightweight, highly concise anonymous methods backed by functional interfaces in Java 8. You can use them to leap forward into a whole new world of programming in Java. With functional programming capabilities, which have been around for decades in other languages, you can now write elegant, concise, less error-prone code using standard Java. This book will guide you though the paradigm change, offer the essential details about the new features, and show you how to transition from your old way of coding to an improved style. In this book you'll see popular design patterns, such as decorator, builder, and strategy, come to life to solve common design problems, but with little ceremony and effort. With these new capabilities in hand, Functional Programming in Java will help you pick up techniques to implement designs that were beyond easy reach in earlier versions of Java. You'll see how you can reap the benefits of tail call optimization, memoization, and effortless parallelization techniques. Java 8 will change the way you write applications. If you're eager to take advantage of the new features in the language, this is the book for you. What you need: Java 8 with support for lambda

expressions and the JDK is required to make use of the concepts and the examples in this book.

Functional Programming in Java

Describes how to use Scala to create applications for the Java VM.

Building Applications with Scala

Write modern, scalable, and reactive applications with the power of Scala About This Book Delves into the intricacies of functional reactive programming with Scala Explores frameworks like Akka, Play and Slick used to develop efficient applications A step by step guide with plenty of examples showing practical implementation of essential concepts Who This Book Is For If you are a Java or JVM developer who wants to use Scala to build reactive functional applications for the JVM platform, then this book is for you. Prior knowledge of Java or functional programing would help. No Scala knowledge is required. What You Will Learn Use Akka to create a chat service for your app Equip yourself with the techniques and tools to build reports and build database persistence with Scala and Slick Develop a customer-facing Rest API that makes use of Scala and Spray Make use of the Scala web development principles and scale up the architecture of your application Get familiar with the core principles and concepts of Functional Programming Use the Play framework to create models, controllers, and views Develop reactive backing frameworks by writing code with RxScala Discover what proper

testing entails with Scala using behavior-driven development In Detail Scala is known for incorporating both object-oriented and functional programming into a concise and extremely powerful package. However, creating an app in Scala can get a little tricky because of the complexity the language has. This book will help you dive straight into app development by creating a real, reactive, and functional application. We will provide you with practical examples and instructions using a hands-on approach that will give you a firm grounding in reactive functional principles. The book will take you through all the fundamentals of app development within Scala as you build an application piece by piece. We've made sure to incorporate everything you need from setting up to building reports and scaling architecture. This book also covers the most useful tools available in the Scala ecosystem, such as Slick, Play, and Akka, and a whole lot more. It will help you unlock the secrets of building your own up-to-date Scala application while maximizing performance and scalability. Style and approach This book takes a step-by-step approach to app development with Scala. It will place special emphasis on functional language. It will teach you the core benefits of Scala and the fundamentals of functional programming by developing a robust application.

Learning Concurrent Programming in Scala

Build domain specific languages (DSLs) using Java's most popular functional programming language:

Scala. This book introduces the basics of Scala and DSLs using a series of practical examples. In Practical Scala DSLs, you'll learn to create pragmatic and complete code examples that explain the actual use of DSLs with Scala: a web API and microservices; a custom language; a mobile app; a Forex system; a game; and cloud applications. At the end of this unique book, you'll be able to describe the differences between external and internal DSLs; understand when and how to apply DSLs; create DSLs using Scala; and even create a DSL using another programming language. What You'll Learn Build DSLs in Scala Write a web API and microservices Create a custom language Apply DSLs to mobile apps development, a Forex trading system, game development, and more Discover the role of DSLs in cloud development Integrate DSLs as part of a DevOps program or structure Build internal and external DSLs Who This Book Is For Experienced Java coders with at least some prior experience with Scala. You may be new to DSLs.

A Beginner's Guide to Scala, Object Orientation and Functional Programming

Get started with Apache Flink, the open source framework that powers some of the world's largest stream processing applications. With this practical book, you'll explore the fundamental concepts of parallel stream processing and discover how this technology differs from traditional batch data processing. Longtime Apache Flink committers Fabian Hueske and Vasia Kalavri show you how to implement

scalable streaming applications with Flink's DataStream API and continuously run and maintain these applications in operational environments. Stream processing is ideal for many use cases, including low-latency ETL, streaming analytics, and real-time dashboards as well as fraud detection, anomaly detection, and alerting. You can process continuous data of any kind, including user interactions, financial transactions, and IoT data, as soon as you generate them. Learn concepts and challenges of distributed stateful stream processing Explore Flink's system architecture, including its event-time processing mode and fault-tolerance model Understand the fundamentals and building blocks of the DataStream API, including its time-based and statefuloperators Read data from and write data to external systems with exactly-once consistency Deploy and configure Flink clusters Operate continuously running streaming applications

Practical Scala DSLs

Our industry is moving toward functional programming, but your object-oriented experience is still valuable. Scala combines the power of OO and functional programming, and Pragmatic Scala shows you how to work effectively with both. Updated to Scala 2.11, with in-depth coverage of new features such as Akka actors, parallel collections, and tail call optimization, this book will show you how to create stellar applications. The first edition of this book was released as Programming Scala. Our industry is moving toward functional programming, but your

object-oriented experience is still valuable. Scala combines the power of OO and functional programming, and Pragmatic Scala shows you how to work effectively with both. Updated to Scala 2.11, with in-depth coverage of new features such as Akka actors, parallel collections, and tail call optimization, this book will show you how to create stellar applications. This thorough introduction to Scala will get you coding in this powerful language right away. You'll start from the familiar ground of Java and, with easy-to-follow examples, you'll learn how to create highly concise and expressive applications with Scala. You'll find out when and how to mix both imperative and functional style, and how to use parallel collections and Akka actors to create high-performance concurrent applications that effectively use multicore processors. Scala has evolved since the first edition of this book, and Pragmatic Scala is a significant update. We've revised each chapter, and added three new chapters and six new sections to explore the new features in Scala. You'll learn how to: Safely manage concurrency with parallel collections and Akka actors Create expressive readable code with value classes and improved implicit conversions Create strings from data with no sweat using string interpolation Create domain-specific languages Optimize your recursions with tail call optimization Whether you're interested in creating concise, robust single-threaded applications or highly expressive, thread-safe concurrent programs, this book has you covered. What You Need: The Scala compiler (2.x) and the JDK are required to make use of the concepts and the examples in this book.

Scala in Depth

Access the power of bare-metal systems programming with Scala Native, an ahead-of-time Scala compiler. Without the baggage of legacy frameworks and virtual machines, Scala Native lets you re-imagine how your programs interact with your operating system. Compile Scala code down to native machine instructions; seamlessly invoke operating system APIs for low-level networking and IO; control pointers, arrays, and other memory management techniques for extreme performance; and enjoy instant start-up times. Skip the JVM and improve your code performance by getting close to the metal. Developers generally build systems on top of the work of those who came before, accumulating layer upon layer of abstraction. Scala Native provides a rare opportunity to remove layers. Without the JVM, Scala Native uses POSIX and ANSI C APIs to build concise, expressive programs that run unusually close to bare metal. Scala Native compiles Scala code down to native machine instructions instead of JVM bytecode. It starts up fast, without the sluggish warm-up phase that's common for just-in-time compilers. Scala Native programs can seamlessly invoke operating system APIs for low-level networking and IO. And Scala Native lets you control pointers, arrays, and other memory layout types for extreme performance. Write practical, bare-metal code with Scala Native, step by step. Understand the foundations of systems programming, including pointers, arrays, strings, and memory management. Use the UNIX socket API to write network client and server programs without the

sort of frameworks higher-level languages rely on. Put all the pieces together to design and implement a modern, asynchronous microservice-style HTTP framework from scratch. Take advantage of Scala Native's clean, modern syntax to write lean, high-performance code without the JVM. What You Need: A modern Windows, Mac OS, or Linux system capable of running Docker. All code examples in the book are designed to run on a portable Docker-based build environment that runs anywhere. If you don't have Docker yet, see the Appendix for instructions on how to get it.

ROMANCE ACTION & ADVENTURE MYSTERY & THRILLER BIOGRAPHIES & HISTORY CHILDREN'S YOUNG ADULT FANTASY HISTORICAL FICTION HORROR LITERARY FICTION NON-FICTION SCIENCE FICTION