

Beginning Haskell A Project Based Approach

The professional programmer's Deitel® guide to Android™ smartphone and tablet app development and the Eclipse IDE with the Android Development Tools (ADT) plug-in Billions of apps have been downloaded from Android Market! This book gives you everything you'll need to start developing great Android apps quickly and getting them published on Android Market. The book uses an app-driven approach—each new technology is discussed in the context of 16 fully tested Android apps, complete with syntax coloring, code walkthroughs and sample outputs. Apps you'll develop include: SpotOn Game Slideshow Fluz Quiz Route Tracker Favorite Twitter@ Searches Address Book Tip Calculator Doodlz Weather Viewer Cannon Game Voice Recorder Pizza Ordering Practical, example-rich coverage of: Smartphone and Tablet Apps, Android Development Tools (ADT) Plug-In for Eclipse Activities, Intents, Content Providers GUI Components, Menus, Toasts, Resource Files, Touch and Gesture Processing Tablet Apps, ActionBar and AppWidgets Tweened Animations, Property Animations Camera, Audio, Video, Graphics, OpenGL ES Gallery and Media Library Access Shared/Preferences, Serialization, SQLite Handlers and Multithreading, Games Google Maps, GPS, Location Services, Sensors Internet-Enabled Apps, Web Services, Telephony, Bluetooth® Speech Synthesis and Recognition Android Market, Pricing, Monetization And more... PLUS: Register your product at www.informit.com/register for additional online chapters that cover Android Ice Cream Sandwich (Android 4), including a complete, working Ice Cream Sandwich app! VISIT WWW.DEITEL.COM For information on Deitel's Dive Into® Series instructor-led programming language training courses offered at customer sites worldwide visit www.deitel.com/training or write to deitel@deitel.com Download code examples Check out the growing list of programming Resource Centers Join the Deitel Twitter (@deitel) and Facebook (www.facebook.com/DeitelFan) communities To receive updates for this book, subscribe to the free Deitel® Buzz Online e-mail newsletter at www.deitel.com/newsletter/subscribe.html

Save time and build fast, functional, and concurrent application using Haskell About This Book Comprehensive guide for establishing a strong foundation in Haskell and developing pragmatic code Create a full fledged web application using Haskell Work with Lens, Haskell Extensions, and write code for concurrent and distributed applications Who This Book Is For This book is targeted at readers who wish to learn the Haskell language. If you are a beginner, Haskell Cookbook will get you started. If you are experienced, it will expand your knowledge base. A basic knowledge of programming will be helpful. What You Will Learn Use functional data structures and algorithms to solve problems Understand the intricacies of the type system Create a simple parser for integer expressions with additions Build high-performance web services with Haskell Master mechanisms for concurrency and parallelism in Haskell Perform parsing and handle scarce resources such as filesystem handles Organize your programs by creating your own types and type classes In Detail Haskell is a purely functional language that has the great ability to develop large and difficult, but easily maintainable software. Haskell Cookbook provides recipes that start by illustrating the principles of functional programming in Haskell, and then gradually build up your expertise in creating industrial-strength programs to accomplish any goal. The book covers topics such as Functors, Applicatives, Monads, and Transformers. You will learn various ways to handle state in your applications and explore advanced topics such as Generalized Algebraic Data Types, Higher kind types, existential types, and type families. The book will discuss the association of lenses with type classes such as Functor, Foldable, and Traversable to help you manage deep data structures. With the help of the wide selection of examples in this book, you will be able to upgrade your Haskell programming skills and develop scalable software idiomatically. Style and approach The book follows a recipe-based approach. Each recipe addresses specific problems and issues. The recipes provide discussions and insights to explain these problems.

Haskell is the world's leading lazy functional programming language, widely used for teaching, research, and applications. The language continues to develop rapidly, but in 1998 the community decided to capture a stable snapshot of the language: Haskell 98. All Haskell compilers support Haskell 98, so practitioners and educators alike have a stable base for their work. This book constitutes the agreed definition of Haskell 98, both the language itself and its supporting libraries, and should be a standard reference work for anyone involved in research, teaching, or application of Haskell. Take your Haskell and functional programming skills to the next level by exploring new idioms and design patterns About This Book Explore Haskell on a higher level through idioms and patterns Get an in-depth look into the three strongholds of Haskell: higher-order functions, the Type system, and Lazy evaluation Expand your understanding of Haskell and functional programming, one line of executable code at a time Who This Book Is For If you're a Haskell programmer with a firm grasp of the basics and ready to move more deeply into modern idiomatic Haskell programming, then this book is for you. What You Will Learn Understand the relationship between the "Gang of Four" OOP Design Patterns and Haskell Try out three ways of Streaming I/O: imperative, Lazy, and Iteratee based Explore the pervasive pattern of Composition: from function composition through to high-level composition with Lenses Synthesize Functor, Applicative, Arrow and Monad in a single conceptual framework Follow the grand arc of Fold and Map on lists all the way to their culmination in Lenses and Generic Programming Get a taste of Type-level programming in Haskell and how this relates to dependently-typed programming Retrace the evolution, one key language extension at a time, of the Haskell Type and Kind systems Place the elements of modern Haskell in a historical framework In Detail Design Patterns and idioms can widen our perspective by showing us where to look, what to look at, and ultimately how to see what we are looking at. At their best, patterns are a shorthand method of communicating better ways to code (writing less, more maintainable, and more efficient code). This book starts with Haskell 98 and through the lens of patterns and idioms investigates the key advances and programming styles that together make "modern Haskell". Your journey begins with the three pillars of Haskell. Then you'll experience the problem with Lazy I/O, together with a solution. You'll also trace the hierarchy formed by Functor, Applicative, Arrow, and Monad. Next you'll explore how Fold and Map are generalized by Foldable and Traversable, which in turn is unified in a broader context by functional Lenses. You'll delve more deeply into the Type system, which will prepare you for an overview of Generic programming. In conclusion you go to the edge of Haskell by investigating the Kind system and how this relates to Dependently-typed programming. Style and approach Using short pieces of executable code, this guide gradually explores the broad pattern landscape of modern Haskell. Ideas are presented in their historical context and arrived at through intuitive derivations, always with a focus on the problems they solve.

Akka in Action

with more than 100 exercises and answers

Proceedings of the ... ACM SIGPLAN Haskell Workshop

Steven Spielberg

Haskell pdf Ebook

Property-Based Testing with PropEr, Erlang, and Elixir

Property-based testing helps you create better, more solid tests with little code. By using the PropEr framework in both Erlang and Elixir, this book teaches you how to automatically generate test cases, test stateful programs, and change how you design your software for more principled and reliable approaches. You will be able to better explore the problem space, validate the assumptions you make when coming up with program behavior, and expose unexpected weaknesses in your design. PropEr will even show you how to reproduce the bugs it found. With this book, you will be writing efficient property-based tests in no time. Most tests only demonstrate that the code behaves how the developer expected it to behave, and therefore carry the same blind spots as their authors when special conditions or edge cases show up. Learn how to see things differently with property tests written in PropEr. Start with the basics of property tests, such as writing stateless properties, and using the default generators to generate test cases automatically. More importantly, learn how to think in properties. Improve your properties, write custom data generators, and discover what your code can or cannot do. Learn when to use property tests and when to stick with example tests with real-world sample projects. Explore various testing approaches to find the one that's best for your code. Shrink failing test cases to their simpler expression to highlight exactly what breaks in your code, and generate highly relevant data through targeted properties. Uncover the trickiest bugs you can think of with nearly no code at all with two special types of properties based on state transitions and finite state machines. Write Erlang and Elixir properties that generate the most effective tests you'll see, whether they are unit tests or complex integration and system tests. What You Need Basic knowledge of Erlang, optionally Elixir For Erlang tests: Erlang/OTP >= 20.0, with Rebar >= 3.4.0 For Elixir tests: Erlang/OTP >= 20.0, Elixir >= 1.5.0

If you have a working knowledge of Haskell, this hands-on book shows you how to use the language's many APIs and frameworks for writing both parallel and concurrent programs. You'll learn how parallelism exploits multicore processors to speed up computation-heavy programs, and how concurrency enables you to write programs with threads for multiple interactions. Author Simon Marlow walks you through the process with lots of code examples that you can run, experiment with, and extend. Divided into separate sections on Parallel and Concurrent Haskell, this book also includes exercises to help you become familiar with the concepts presented. Express parallelism in Haskell with the Eval monad and Evaluation Strategies Parallelize ordinary Haskell code with the Par monad Build parallel array-based computations using the Reps library Use the Accelerate library to run computations directly on the GPU Work with basic interfaces for writing concurrent code Build trees of threads for larger and more complex programs Learn how to build high-speed concurrent network servers Write distributed programs that run on multiple machines in a network.

Summary Functional Programming in Scala is a serious tutorial for programmers looking to learn FP and apply it to the everyday business of coding. The book guides readers from basic techniques to advanced topics in a logical, concise, and clear progression. In it, you'll find concrete examples and exercises that open up the world of functional programming. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Functional programming (FP) is a style of software development emphasizing functions that don't depend on program state. Functional code is easier to test and reuse, simpler to parallelize, and less prone to bugs than other code. Scala is an emerging JVM language that offers strong support for FP. Its familiar syntax and transparent interoperability with Java make Scala a great place to start learning FP. About the Book Functional Programming in Scala is a serious tutorial for programmers looking to learn FP and apply it to their everyday work. The book guides readers from basic techniques to advanced topics in a logical, concise, and clear progression. In it, you'll find concrete examples and exercises that open up the world of functional programming. This book assumes no prior experience with functional programming. Some prior exposure to Scala or Java is helpful. What's Inside Functional programming concepts The whys and hows of FP How to write multicore programs Exercises and checks for understanding About the Authors Paul Chiusano and Rúnar Bjarnason are recognized experts in functional programming with Scala and are core contributors to the Scalaz library. Table of Contents PART 1 INTRODUCTION TO FUNCTIONAL PROGRAMMING What is functional programming? Getting started with functional programming in Scala Functional data structures Handling errors without exceptions Strictness and laziness Purely functional state PART 2 FUNCTIONAL DESIGN AND COMBINATOR LIBRARIES Purely functional parallelism Property-based testing Parser combinators PART 3 COMMON STRUCTURES IN FUNCTIONAL DESIGN Monoids Monads Applicative and traversable functors PART 4 EFFECTS AND I/O External effects and I/O Local effects and mutable state Stream processing and incremental I/O

It's all in the name: Learn You a Haskell for Great Good! is a hilarious, illustrated guide to this complex functional language. Packed with the author's original artwork, pop culture references, and most importantly, useful example code, this book teaches functional fundamentals in a way you never thought possible. You'll start with the kid stuff: basic syntax, recursion, types and type classes. Then once you've got the basics down, the real black belt master-class begins: you'll learn to use applicative functors, monads, zippers, and all the other mythical Haskell constructs you've only read about in storybooks. As you work your way through the author's imaginative (and occasionally insane) examples, you'll learn to: -Laugh in the face of side effects as you wield purely functional programming techniques -Use the magic of Haskell's "laziness" to play with infinite sets of data -Organize your programs by creating your own types, type classes, and modules -Use Haskell's elegant input/output system to share the genius of your programs with the outside world Short of eating the author's brain, you will not find a better way to learn this powerful language than reading Learn You a Haskell for Great Good!

Recent Research in Control Engineering and Decision Making

A Hands-On, Project-Based Introduction to Programming

Haskell in Depth

Haskell Programming from First Principles

Haskell 98 Language and Libraries

The second edition of the best-selling Python book in the world (over 1 million copies sold!). A fast-paced, no-nonsense guide to programming in Python. Updated and thoroughly revised to reflect the latest in Python code and practices. Python Crash Course is the world's best-selling guide to the Python programming language. This fast-paced, thorough introduction to programming with Python will have you writing programs, solving problems, and making things that work in no time. In the first half of the book, you'll learn basic programming concepts, such as variables, lists, classes, and loops, and practice writing clean code with exercises for each topic. You'll also learn how to make your programs interactive and test your code safely before adding it to a project. In the second half, you'll put your new knowledge into practice with three substantial projects: a web browser, a data visualization, and a game. You'll also learn how to use powerful Python libraries and tools, including Pygame, Matplotlib, Plotty, and Django • Make 2D games that respond to keypresses and mouse clicks, and that increase in difficulty • Use data to generate interactive visualizations • Create and customize web apps and deploy them safely online • Deal with mistakes and errors so you can solve your own programming problems If you've been thinking about digging into programming, Python Crash Course will get you writing real programs fast. Why wait any longer? Start your engines and code!

2012 Jolt Award Finalist! Even experienced software professionals find it difficult to apply patterns in ways that deliver substantial value to their organizations. In Elemental Design Patterns, Jason M.C. Smith addresses this problem head-on, helping developers harness the true power of patterns, map them to real software implementations more cleanly and directly, and achieve far better results. Part tutorial, part example-rich cookbook, this resource will help developers, designers, architects, and analysts successfully use patterns with a wide variety of languages, environments, and problem domains. Every bit as important, it will give them a deeper appreciation for the work they've chosen to pursue. Smith presents the crucial missing link that patterns practitioners have needed: a foundational collection of simple core patterns that are broken down to their core elements. If you work in software, you may already be using some of these elemental design patterns every day. Presenting them in a comprehensive methodology for the first time, Smith names them, describes them, explains their importance, helps you compare and choose among them, and offers a framework for using them together. He also introduces an innovative Pattern Instance Notation diagramming system that makes it easier to work with patterns at many levels of granularity, regardless of your goals or role. If you're new to patterns, this example-rich approach will help you master them piece by piece, logically and intuitively. If you're an experienced patterns practitioner, Smith follows the Gang of Four format you're already familiar with, explains how his elemental patterns can be composed into conventional design patterns, and introduces highly productive new ways to apply ideas you've already encountered. No matter what your level of experience, this infinitely practical book will help you transform abstract patterns into high-value solutions.

Introduces fundamles for reasoning mathematically about functional programs. Ideal for a first- or second-year undergraduate course. Summary Get Programming with F#: A guide for .NET developers teaches F# through 43 example-based lessons with built-in exercises so you can learn the only way that really works: by practicing. The book upgrades your .NET skills with a touch of functional programming in F#. You'll pick up core FP principles and learn techniques for iron-clad reliability and crystal clarity. You'll discover productivity techniques for coding F# in Visual Studio, functional design, and integrating functional and OO code. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Your .NET applications need to be good for the long haul. F#'s unique blend of functional and imperative programming is perfect for writing code that performs flawlessly now and keeps running as your needs grow and change. It takes a little practice to master F#'s functional-first style, so you may as well get programming! What's Inside Learn how to write bug-free programs Turn tedious common tasks into quick and easy ones Use minimal code to work with JSON, CSV, XML, and HTML data Integrate F# with your existing C# and VB.NET applications Create web-enabled applications About the Reader Written for intermediate C# and Visual Basic .NET developers. No experience with F# is assumed. Table of Contents Unit 1 - F# AND VISUAL STUDIO Lesson 1 - The Visual Studio experience Lesson 2 - Creating your first F# program Lesson 3 - The REPL-changing how we develop Unit 2 - HELLO F# Lesson 4 - Saying a little, doing a lot Lesson 5 - Trusting the compiler Lesson 6 - Working with immutable data Lesson 7 - Expressions and statements Lesson 8 Capstone 1 Unit 3 - TYPES AND FUNCTIONS Lesson 9 - Shaping data with tuples Lesson 10 - Shaping data with records Lesson 11 - Building composable functions Lesson 12 - Organizing code without classes Lesson 13 - Achieving code reuse in F# Lesson 14 - Capstone 2 Unit 4 - COLLECTIONS IN F# Lesson 15 - Working with collections in F# Lesson 16 - Useful collection functions Lesson 17 - Maps, dictionaries, and sets Lesson 18 - Folding your way to success Lesson 19 - Capstone 3 Unit 5 - THE PIT OF SUCCESS WITH THE F# TYPE SYSTEM Lesson 20 - Program Flow in F# Lesson 21 - Modeling relationships in F# Lesson 22 - Fixing the billion-dollar mistake Lesson 23 - Business rules as code Lesson 24 - Capstone 4 Unit 6 - LIVING ON THE .NET PLATFORM Lesson 25 - Consuming C# from F# Lesson 26 - Working with NuGet packages Lesson 27 - Exposing F# types and functions C# Lesson 28 - Architecting hybrid language applications Lesson 29 - Capstone 5 Unit 7 - WORKING WITH DATA Lesson 30 - Introducing type providers Lesson 31 - Building schemas from live data Lesson 32 - Working with SQL Lesson 33 - Creating type provider-backed APIs Lesson 34 - Using type providers in the real world Lesson 35 - Capstone 6 Unit 8 - WEB PROGRAMMING Lesson 36 - Asynchronous workflows Lesson 37 - Exposing data over HTTP Lesson 38 - Consuming HTTP data Lesson 39 - Capstone 7 Unit 9 - UNIT TESTING Lesson 40 - Unit testing in F# Lesson 41 - Property-based testing in F# Lesson 42 - Web testing Lesson 43 - Capstone 8 Unit 10 - WHERE NEXT? Appendix A - The F# community Appendix B - F# in my organization Appendix C - Must-visit F# resources Appendix D - Must-have F# libraries APPENDIX E - Other F# language feature

Python Crash Course, 2nd Edition

Get Programming with Haskell

Practical Haskell

Build functional applications using Monads, Applicatives, and Functors

Haskell Cookbook

Thinking Functionally with Haskell

Learn to use the APIs and frameworks for parallel and concurrent applications in Haskell. This book will show you how to exploit multicore processors with the help of parallelism in order to increase the performance of your applications. Practical Concurrent Haskell teaches you how concurrency enables you to write programs using threads for multiple interactions. After accomplishing this, you will be ready to make your move into application development and portability with applications in cloud computing and big data. You'll use MapReduce and other, similar big data tools as part of your Haskell big data applications development. What You'll Learn Program with Haskell Harness concurrency to Haskell Apply Haskell to big data and cloud computing applications Use Haskell concurrency design patterns in big data Accomplish iterative data processing on big data using Haskell Use MapReduce and work with Haskell on large clusters Who This Book Is For Those with at least some prior experience with Haskell and some prior experience with big data in another programming language such as Java, C#, Python, or C++ WINNER OF THE 2018 JOHN BURROUGHS MEDAL FOR OUTSTANDING NATURAL HISTORY WRITING "Both a love song to trees, an exploration of their biology, and a wonderfully philosophical analysis of their role they play in human history and in modern culture."—Science Friday The author of Sounds Wild and Broken and the Pulitzer Prize finalist The Forest Unseen visits with nature's most magnificent networkers—trees David Haskell has won acclaim for eloquent writing and deep engagement with the natural world. Now, he brings his powers of observation to the biological networks that surround all species, including humans. Haskell repeatedly visits a dozen trees, exploring connections with people, microbes, fungi, and other plants and animals. He takes us to trees in cities (from Manhattan to Jerusalem), forests (Amazonian, North American, and boreal) and areas on the front lines of environmental change (eroding coastlines, burned mountainsides, and war zones.) In each place he shows how human history, ecology, and well-being are intimately intertwined with the lives of trees. Scientific, lyrical, and contemplative, Haskell reveals the biological connections that underpin all life. In a world beset by barriers, he reminds us that life's sustenance and beauty emerge from relationship and interdependence.

Summary Get Programming with Haskell leads you through short lessons, examples, and exercises designed to make Haskell your own. It has crystal-clear illustrations and guided practice. You will write and test dozens of interesting programs and dive into custom Haskell modules. You will gain a new perspective on programming plus the practical ability to use Haskell in the everyday world. (The 80 IQ points: not guaranteed.) Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Programming languages often differ only around the edges—a few keywords, libraries, or platform choices. Haskell gives you an entirely new point of view. To the software pioneer Alan Kay, a change in perspective can be worth 80 IQ points and Hacklers agree on the dramatic benefits of thinking the Haskell way—thinking functionally, with type safety, mathematical certainty, and more. In this hands-on book, that's exactly what you'll learn to do. What's Inside Thinking in Haskell Functional programming basics Programming in types Real-world applications for Haskell About the Reader Written for readers who know one or more programming languages. Table of Contents Lesson 1 Getting started with Haskell Unit 1 - FOUNDATIONS OF FUNCTIONAL PROGRAMMING Lesson 2 Functions and functional programming Lesson 3 Lambda functions and lexical scope Lesson 4 First-class functions Lesson 5 Closures and partial application Lesson 6 Lists Lesson 7 Rules for recursion and pattern matching Lesson 8 Writing recursive functions Lesson 9 Higher-order functions Lesson 10 Capstone: Functional object-oriented programming with robots Unit 2 - INTRODUCING TYPES Lesson 11 Type basics Lesson 12 Creating your own types Lesson 13 Type classes Lesson 14 Using type classes Lesson 15 Capstone: Secret messages! Unit 3 - PROGRAMMING IN TYPES Lesson 16 Creating types with "and" and "or" Lesson 17 Design by composition—Semigroups and Monoids Lesson 18 Parameterized types Lesson 19 The Maybe type Lesson 20 Capstone: Time series Unit 4 - IO IN HASSELL Lesson 21 Hello, introducing IO types Lesson 22 Interacting with the command line and lazy IO Lesson 23 Working with text and Unicode Lesson 24 Working with files Lesson 25 Working with binary data Lesson 26 Capstone: Processing binary files and book data Unit 5 - WORKING WITH TYPE IN A CONTEXT Lesson 27 The Functor type class Lesson 28 A peek at the Applicative type class: using functions in a context Lesson 29 Lists as context: a deeper look at the Applicative type class Lesson 30 Introducing the Monad type class Lesson 31 Making Monads easier with notation Lesson 32 The list monad and list comprehensions Lesson 33 Capstone: SQL-like queries in Haskell Unit 6 - ORGANIZING CODE AND BUILDING PROJECTS Lesson 34 Organizing Haskell code with modules Lesson 35 Building projects with stack Lesson 36 Property testing with QuickCheck Lesson 37 Capstone: Building a prime-number library Unit 7 - PRACTICAL HASSELL Lesson 38 Errors in Haskell and the Either type Lesson 39 Making HTTP requests in Haskell Lesson 40 Working with JSON data by using Aeson Lesson 41 Using databases in Haskell Lesson 42 Efficient, stateful arrays in Haskell Afterword - What's next? Appendix - Sample answers to exercise

This condensed code and syntax reference presents the essential Haskell syntax in a well-organized format that can be used as a quick and handy reference, including applications to cloud computing and data analysis. This book covers the functional programming features of Haskell as well as strong static typing, lazy evaluation, extensive parallelism, and concurrency You won't find any technical jargon, bloated syntax, or drawn-out lengthy lessons. Every entry in this book, what you will find is a language reference that is concise, to the point and highly accessible. The Haskell Quick Syntax Reference is packed with useful information and is a must-have for any Haskell programmer working in big data, data science, and cloud computing. What You Will Learn Quickly and effectively use the Haskell programming language Take advantage of strong static typing Work with lazy evaluations Harness concurrency and extensive parallelism using Haskell Who This Book Is For Experienced programmers who may be new to Haskell or have experience with Haskell and who just want a quick reference guide on it.

A Beginner's Guide

Beginning Haskell

Programming in Haskell

Python from the Very Beginning

Real World Haskell

Design of Embedded Systems Using 68HC12/11 Microcontrollers

Design patterns and architectures for building production quality applications using functional programming, with examples in Haskell and other FP languages. Functional Design and Architecture is a comprehensive guide to software engineering using functional programming. Inside, you'll find cutting-edge functional design principles and practices for every stage of application development. There's no abstract theory—you'll learn by building exciting sample applications, including an application for controlling a spaceship and a full-fledged backend framework. You'll explore functional design by looking at object-oriented principles you might already know, and learn how they can be reapplied to a functional environment. By the time you're done, you'll be ready to apply the brilliant innovations of the functional world to serious software projects.

Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. Haskell is one of the leading languages for teaching functional programming, enabling students to write simpler and cleaner code, and to learn how to structure and reason about programs. This introduction is ideal for beginners: it requires no previous programming experience and all concepts are explained from first principles via carefully chosen examples. Each chapter includes exercises that range from the straightforward to extended projects, plus suggestions for further reading on more advanced topics. The author is a leading Haskell researcher and instructor, well-known for his teaching skills. The presentation is clear and simple, and benefits from having been refined and class-tested over several years. The result is a text that can be used with courses, or for self-learning. Features include freely accessible Powerpoint slides for each chapter, solutions to exercises and examination questions (with solutions) available to instructors, and a downloadable code that's fully compliant with the latest Haskell release.

Place of publication taken from publisher's web site.

In Python from the Very Beginning John Whittington takes a no-prerequisites approach to teaching the basics of a modern general-purpose programming language. Each small, self-contained chapter introduces a new topic, building until the reader can write quite substantial programs. There are plenty of questions and, crucially, worked answers and hints. Python from the Very Beginning will appeal both to new programmers, and to experienced programmers eager to explore functional languages such as Haskell. It is suitable both for formal use within an undergraduate or graduate curriculum, and for the interested amateur.

Conceptual Programming with Python

Haskell Design Patterns

The Songs of Trees

Learn You a Haskell for Great Good!

Techniques for Multicore and Multithreaded Programming

Haskell Quick Syntax Reference

Introducing functional programming in the Haskell language, this book is written for students and programmers with little or no experience. It emphasizes the process of crafting programmes, problem solving and avoiding common programming pitfalls. Covering basic functional programming, through abstraction to larger scale programming, students are lead step by step through the basics, before being introduced to more advanced topics. This edition includes new material on testing and domain-specific languages and a variety of new examples and case studies, including simple games. Existing material has been expanded and re-ordered, so that some concepts - such as simple data types and input/output - are presented at an earlier stage. The full text downloaded to your computer With eBooks you can: search for key concepts, words and phrases make highlights and notes as you study share your notes with friends eBooks are downloaded to your computer and accessible either offline through the Bookshelf (available as a free download), available online and also via the iPad and Android apps. Upon purchase, you'll gain instant access to this eBook. Time limit The eBooks products do not have an expiry date. You will continue to access your digital eBook products whilst you have your Bookshelf installed.

This book constitutes the full papers and short monographs developed on the base of the refereed proceedings of the International Conference on Information Technologies: Information and Communication Technologies for Research and Industry (ICIT-2019), held in Saratov, Russia in February 2019. The book brings accepted papers which present new approaches and methods of solving problems in the sphere of control engineering and decision making for the various fields of studies: industry and research, ontology-based data simulation, smart city technologies, theory and use of digital signal processing, cognitive systems, robotics, cybernetics, automation control theory, image recognition technologies, and computer vision. Particular emphasis is laid on modern trends, new approaches, algorithms and methods in selected fields of interest. The book is presented as accepted careful reviews made by at least three independent reviewers using a double-blind way. The acceptance level was about 60%. The chapters are organized thematically in several areas within the following tracks: • Models, Methods & Approaches in Decision Making Systems • Mathematical Modelling for Industry & Research • Smart City Technologies The conference is focused on development and globalization of information and communication technologies (ICT), methods of control engineering and decision making along with innovations and networking, ICT for sustainable development and technological change, and global challenges. Moreover, the ICIT-2019 served as a discussion area for the actual above-mentioned topics. The editors believe that the readers will find the proceedings interesting and useful for their own research work.

Summary Akka in Action is a comprehensive tutorial on building message-oriented systems using Akka. The book takes a hands-on approach, where each new concept is followed by an example that shows you how it works, how to implement the code, and how to (unit) test it. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Akka makes it relatively easy to build applications in the cloud or on devices with many cores that efficiently use the full capacity of the computing power available. It's a toolkit that provides an actor programming model, a runtime, and required support tools for building scalable applications. About the Book Akka in Action shows you how to build message-oriented systems with Akka. This comprehensive, hands-on tutorial introduces each concept with a working example. You'll start with the big picture of how Akka works, and then quickly build and deploy a fully functional REST service out of actors. You'll explore test-driven development and deploying and scaling fault-tolerant systems. After mastering the basics, you'll discover how to model immutable messages, implement domain models, and apply techniques like event sourcing and CQRS. You'll also learn how to build streaming applications using akka-stream and akka-http. Finally, you'll get practical advice on how to customize and extend your Akka system. What's Inside Getting concurrency right Testing and performance tuning Clustered and cloud-based applications Covers Akka version 2.4 About the Reader This book assumes that you're comfortable with Java and Scala. No prior experience with Akka required. About the Authors A software craftsman and architect, Raymond Roestenburg is an Akka committer. Rob Bakker specializes in concurrent back-end systems and systems integration. Rob Williams has more than 20 years of product development experience. Table of Contents Introducing Akka Up and running Test-driven development with actors Fault tolerance Futures Your first distributed Akka app Configuration, logging, and deployment Structural patterns for actors Routing messages Message channels Finite-state machines and agents System integration Streaming Clustering Actor persistence Performance tips Looking ahead

Beginning HaskellA Project-Based ApproachApress

A guide for .NET developers

An App-Driven Approach

The Revised Report

Functional Programming in Scala

Android for Programmers

The Haskell School of Music

In Haskell from the Very Beginning John Whittington takes a no-prerequisites approach to teaching the basics of a modern general-purpose programming language. Each small, self-contained chapter introduces a new topic, building until the reader can write quite substantial programs. There are plenty of questions and, crucially, worked answers and hints. Haskell from the Very Beginning will appeal both to new programmers, and to experienced programmers eager to explore functional languages such as Haskell. It is suitable both for formal use within an undergraduate or graduate curriculum, and for the interested amateur.

Learn functional programming and the Haskell programming language through algorithmic music composition and virtual instrument design.

This is the first book to describe, in detail, the new Motorola 68HC12 microcontroller, how to program it, and how to design embedded systems using the 68HC12. It shows how WHYP (a version of Forth written specifically for this book) can be used to program the new 68HC12 microcontroller in an efficient and interactive way. It includes an abundance of worked examples and complete C++ code for the WHYP host that runs on the PC. Subroutines and Stacks. 68HC12 Arithmetic, WHYP-An Extensible Language, Branching and Looping, Parallel Interfacing, The Serial Peripheral Interface (SPI), Analog-to-Digital Converter, Timers, The Serial Communications Interface (SCI), Designing with Interrupts, Strings and Number Conversions, Program Control and Data Structures, Fuzzy Control, Special Topics, WHYP12 C++ Classes, WHYP12 C++ Main Program. For electrical and computer engineers who want to learn about the new Motorola 68HC12 microcontroller, how to program it, and how to design embedded systems using it.

A film-centric portrait of the extraordinarily gifted movie director whose decades-long influence on American popular culture is unprecedented Everything about me is in my films, Steven Spielberg has said. Taking this as a key to understanding the hugely successful moviemaker, Molly Haskell explores the full range of Spielberg's works for the light they shine upon the man himself. Through such powerhouse hits as Close Encounters of the Third Kind, E.T., Jurassic Park, and Indiana Jones, to lesser-known masterworks like A.I. and Empire of the Sun, to the haunting Schindler's List, Haskell shows how Spielberg's uniquely evocative filmmaking and story-telling reveal the many ways in which his life, work, and times are entwined. Organizing chapters around specific films, the distinguished critic discusses how Spielberg's childhood in non-Jewish suburbs, his parents' traumatic divorce, his return to Judaism upon his son's birth, and other events echo in his work. She offers a brilliant portrait of the extraordinary director a fearful boy living through his imagination who grew into a man whose openness, generosity of spirit, and creativity have enchanted audiences for more than 40 years.

Four Books Before You're Do

Elemental Design Patterns

A Real World Guide to Programming

The Craft of Functional Programming

Rust Functional Programming in C++

The Rust Programming Language (Covers Rust 2018)

Python Crash Course is a fast-paced, thorough introduction to Python that will have you writing programs, solving problems, and making things that work in no time. In the first half of the book, you'll learn about basic programming concepts, such as lists, dictionaries, classes, and loops, and practice writing clean and readable code with exercises for each topic. You'll also learn how to make your programs interactive and how to test your code safely before adding it to a project. In the second half of the book, you'll put your new knowledge into practice with three substantial projects: a Space Invaders-inspired arcade game, data visualizations with Python's super-handly libraries, and a simple web app you can deploy online. As you work through Python Crash Course you'll learn how to—Use powerful Python libraries and tools, including matplotlib, NumPy, and Pygal -Make 2D games that respond to keypresses and mouse clicks, and that grow more complex as you play -Use data to generate interactive visualizations -Create and customize Web apps and deploy them safely online -Deal with mistakes and errors so you can solve your own programming problems If you've been thinking seriously about digging into programming, Python Crash Course will get you up to speed and have you writing real programs fast. Why wait any longer? Start your engines and code! Uses Python 2 and 3 This is the Scala edition Category Theory for Programmers by Bartosz Milewski. This book contains code snippets in both Haskell and Scala. This easy-to-use, fast-moving tutorial introduces you to functional programming with Haskell. You'll learn how to use Haskell in a variety of practical ways, from short scripts to large and demanding applications. Real World Haskell takes you through the basics of functional programming at a brisk pace, and then helps you increase your understanding of Haskell in real-world issues like I/O, performance, dealing with data, concurrency, and more You may work through each chapter.

Hands-on Scala teaches you how to use the Scala programming language in a practical, project-based fashion. This book is designed to quickly teach an existing programmer everything needed to go from "hello world" to building production applications like interactive websites, parallel web crawlers, and distributed systems in Scala. In the process you will learn how to use the Scala language to solve challenging problems in an elegant and intuitive manner.

Solutions and Examples for Java Developers

Haskell from the Very Beginning

Category Theory for Programmers (Scala Edition, Paperback)

Code You Can Believe In

Get Programming with F#

With Big Data Applications

Thorsten and Isaac have written this book based on the programming course we teach for Master's Students at the School of Computer Science of the University of Nottingham. The book is intended for students with little or no background in programming coming from different backgrounds educationally as well as culturally. It is not mainly programming concepts. Hence, the words conceptual programming in the title. We cover basic concepts about data structures, imperative programming, recursion and backtracking, object-oriented programming, functional programming, game development and some basics of data science.

Get a practical, hands-on introduction to the Haskell language, its libraries and environment, and to the functional programming paradigm that is fast growing in importance in the software industry. This book contains excellent coverage of the Haskell ecosystem and supporting tools, include Cabal and Stack for managing projects, HUnit and developing web applications, Persistent and Esqueleto for database access, and parallel and distributed programming libraries. You'll see how functional programming is gathering momentum, allowing you to express yourself in a more concise way, reducing boilerplate, and increasing the safety of your code. Haskell is an elegant and noise-free huge number of library contributors and an active community. This makes Haskell the best tool for both learning and applying functional programming, and Practical Haskell takes advantage of this to show off the language and what it can do. What You Will Learn Get started programming with Haskell Examine the different parts of the language and how they fit together Apply functional patterns in real-world scenarios Understand monads and monad transformers Proficiently use laziness and resource management Who This Book Is For Experienced programmers who may be new to the Haskell programming language. However, some prior exposure to Haskell is recommended.

Binning Haskell provides a broad-based introduction to the Haskell language, its libraries and environment, and to the functional programming paradigm that is fast growing in importance in the software industry. The book takes a project-based approach to learning the language that is unified around the building of a web-based storage supporting tools. These include the Cabal build tool for managing projects and modules, the HUnit and QuickCheck tools for software testing, the Scotty framework for developing web applications, Persistent and Esqueleto for database access, and also parallel and distributed programming libraries. Functional programming is gathering momentum, allowing you to express yourself in a more concise way, reducing boilerplate and increasing the safety of code. Indeed, mainstream languages such as C# and Java are adopting features from functional programming, and from languages implementing that paradigm. Haskell is an elegant and noise-free pure functional language with a long history, having a huge number of library contributors and an active community. This makes Haskell the best tool for both learning and applying functional programming, and Beginning Haskell the perfect book to show off the language and what it can do. Takes you through a series of projects showing the different parts of the language. Provides an overview of the most important libraries and tools in the Haskell ecosystem. Teaches you how to

Haskell in Depth unlocks a new level of skill with this challenging language. Going beyond the basics of syntax and structure, this book opens up critical topics like advanced types, concurrency, and data processing. Summary Turn the corner from "Haskell student" to "Haskell developer." Haskell in Depth explores the important language features for writing high-quality software using Haskell. And along the way, you'll pick up some interesting insights into why Haskell looks and works the way it does. Get ready to go deep! Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Software for high-precision tasks like financial modeling, absolutely, provably correct. As a purely functional programming language, Haskell enforces a mathematically rigorous approach that can lead to concise, efficient, and bug-free code. To write such code you'll need deep understanding. You can get it from this book! About the book Haskell in Depth unlocks a new level of skill with this challenging language. Going beyond the basics of syntax and structure, this book opens up critical topics like advanced types, concurrency, and data processing. You'll discover key parts of the Haskell ecosystem and master core design patterns that will transform how you write software. What's inside Building applications, web services, and networking apps Using sophisticated libraries like lens, sirius, and more Error-handling and testing Pure parallelism for multicore processors About the reader For developers familiar with Haskell basics. About the author Vitaly Bragilevsky has been teaching Haskell and functional programming since 2008. He is a member of the GHC Steering Committee. Table of Contents PART 1 CORE HASKELL 1 Functions and data types 2 Type system advances 12 Metaprogramming in Haskell 13 More about types PART 5 HASKELL TOOLKIT 14 Data-processing pipelines 15 Working with relational databases 16 Concurrency

From Signals to Symphonies
A Pocket Guide to the Language, APIs, and Library
Stories from Nature's Great Connectors
Practical Concurrent Haskell
Hands-on Scala Programming: Learn Scala in a Practical, Project-Based Way
Python Crash Course
Haskell Programming makes Haskell as clear, painless, and practical as it can be, whether you're a beginner or an experienced hacker. Learning Haskell from the ground up is easier and works better. With our exercise-driven approach, you'll build on previous chapters such that by the time you reach the notorious Monad, it'll seem trivial. The official book on the Rust programming language, written by the Rust development team at the Mozilla Foundation, fully updated for Rust 2018. The Rust Programming Language is the official book on Rust: an open source systems programming language that helps you write faster, more reliable software. Rust offers control over low-level details (such as memory usage) in combination with high-level ergonomics, eliminating the hassle traditionally associated with low-level languages. The authors of The Rust Programming Language, members of the Rust Core Team, share their knowledge and experience to show you how to take full advantage of Rust's features--from installation to creating robust and scalable programs. You'll begin with basics like creating functions, choosing data types, and binding variables and then move on to more advanced concepts, such as: • Ownership and borrowing, lifetimes, and traits • Using Rust's memory safety guarantees to build fast, safe programs • Testing, error handling, and effective refactoring • Generics, smart pointers, multithreading, trait objects, and advanced pattern matching • Using Cargo, Rust's built-in package manager, to build, test, and document your code and manage dependencies • How best to use Rust's advanced compiler with compiler-led programming techniques You'll find plenty of code examples throughout the book, as well as three chapters dedicated to building complete projects to test your learning: a number guessing game, a Rust implementation of a command line tool, and a multithreaded server. New to this edition: An extended section on Rust macros, an expanded chapter on modules, and appendices on Rust development tools and editions.

Summary Functional Programming in C++ teaches developers the practical side of functional programming and the tools that C++ provides to develop software in the functional style. This in-depth guide is full of useful diagrams that help you understand FP concepts and begin to think functionally. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Well-written code is easier to test and reuse, simpler to parallelize, and less error-prone. Mastering the functional style of programming can help you tackle the demands of modern apps and will lead to simpler expression of complex program logic, graceful error handling, and elegant concurrency. C++ supports FP with templates, lambdas, and other core language features, along with many parts of the STL. About the Book Functional Programming in C++ helps you unleash the functional side of your brain, as you gain a powerful new perspective on C++ coding. You'll discover dozens of examples, diagrams, and illustrations that break down the functional concepts you can apply in C++, including lazy evaluation, function objects and invocables, algebraic data types, and more. As you read, you'll match FP techniques with practical scenarios where they offer the most benefit. What's inside Writing safer code with no performance penalties Explicitly handling errors through the type system Extending C++ with new control structures Composing tasks with DSLs About the Reader Written for developers with two or more years of experience coding in C++. About the Author Ivan Čukić is a core developer at KDE and has been coding in C++ since 1998. He teaches modern C++ and functional programming at the Faculty of Mathematics at the University of Belgrade. Table of Contents Introduction to functional programming Getting started with functional programming Function objects Creating new functions from the old ones Purity: Avoiding mutable state Lazy evaluation Ranges Functional data structures Algebraic data types and pattern matching Monads Template metaprogramming Functional design for concurrent systems Testing and debugging From lambda expressions and JavaFX 8 to new support for network programming and mobile development. Java 8 brings a wealth of changes. This cookbook helps you get up to speed right away with hundreds of hands-on recipes across a broad range of Java topics. You'll learn useful techniques for everything from debugging and data structures to GUI development and functional programming. Each recipe includes self-contained code solutions that you can freely use, along with a discussion of how and why they work. If you are familiar with Java basics, this cookbook will bolster your knowledge of the language in general and Java 8's main APIs in particular. Recipes include: Methods for compiling, running, and debugging Manipulating, comparing, and rearranging text Regular expressions for string- and pattern-matching Handling numbers, dates, and times Structuring data with collections, arrays, and other types Object-oriented and functional programming techniques Directory and filesystem operations Working with graphics, audio, and video GUI development, including JavaFX and handlers Network programming on both client and server Database access, using JPA, Hibernate, and JDBC Processing JSON and XML for data storage Multithreading and concurrency

Java Cookbook
Functional Design and Architecture
A Life in Films
A Project-Based Approach
Parallel and Concurrent Programming in Haskell