1. Record Nr. UNINA9910148734903321

| | |
|---|---|
| Autore | Vivien Vladimir |
| Titolo | Learning Go programming : an insightful guide to learning the Go programming language / / Vladimir Vivien |
| Pubbl/distr/stampa | Birmingham, England ; ; Mumbai, India : , : Packt Publishing, , 2016<br>©2016 |
| ISBN | 1-78439-233-2 |
| Edizione | [1st edition] |
| Descrizione fisica | 1 online resource (340 pages) : illustrations |
| Disciplina | 005.133 |
| Soggetti | Go (Computer program language)<br>Computer programming |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | Includes index. |
| Nota di contenuto | Cover -- Copyright -- Credits -- About the Author -- About the Reviewers -- www.PacktPub.com -- Table of Contents -- Preface -- Chapter 1: A First Step in Go -- The Go programming language -- Playing with Go -- No IDE required -- Installing Go -- Source code examples -- Your first Go program -- Go in a nutshell -- Functions -- Packages -- The workspace -- Strongly typed -- Composite types -- The named type -- Methods and objects -- Interfaces -- Concurrency and channels -- Memory management and safety -- Fast compilation -- Testing and code coverage -- Documentation -- An extensive library -- The Go Toolchain -- Summary -- Chapter 2: Go Language Essentials -- The Go source file -- Optional semicolon -- Multiple lines -- Go identifiers -- The blank identifier -- Muting package imports -- Muting unwanted function results -- Built-in identifiers -- Types -- Values -- Functions -- Go variables -- Variable declaration -- The zero-value -- Initialized declaration -- Omitting variable types -- Short variable declaration -- Restrictions for short variable declaration -- Variable scope and visibility -- Variable declaration block -- Go constants -- Constant literals -- Typed constants -- Untyped constants -- Assigning untyped constants -- Constant declaration block -- Constant enumeration -- Overriding the default enumeration type -- Using iota in expressions -- Skipping enumerated values -- Go operators -- Arithmetic operators -- The increment and decrement |

operators -- Go assignment operators -- Bitwise operators -- Logical Operators -- Comparison operators -- Operator precedence -- Summary -- Chapter 3: Go Control Flow -- The if statement -- The if statement initialization -- Switch statements -- Using expression switches -- The fallthrough cases -- Expressionless switches -- Switch initializer -- Type switches -- The for statements -- For condition. Infinite loop -- The traditional for statement -- The for range -- The break, continue, and goto statements -- The label identifier -- The break statement -- The continue statement -- The goto statement -- Summary -- Chapter 4: Data Types -- Go types -- Numeric types -- Unsigned integer types -- Signed integer types -- Floating point types -- Complex number types -- Numeric literals -- Boolean type -- Rune and string types -- The rune -- The string -- Interpreted and raw string literals -- Pointers -- The pointer type -- The address operator -- The new() function -- Pointer indirection - accessing referenced values -- Type declaration -- Type conversion -- Summary -- Chapter 5: Functions in Go -- Go functions -- Function declaration -- The function type -- Variadic parameters -- Function result parameters -- Named result parameters -- Passing parameter values -- Achieving pass-by-reference -- Anonymous Functions and Closures -- Invoking anonymous function literals -- Closures -- Higher-order functions -- Error signaling and handling -- Signaling errors -- Error handling -- The error type -- Deferring function calls -- Using defer -- Function panic and recovery -- Function panic -- Function panic recovery -- Summary -- Chapter 6: Go Packages and Programs -- The Go package -- Understanding the Go package -- The workspace -- Creating a workspace -- The import path -- Creating packages -- Declaring the package -- Multi-File packages -- Naming packages -- Use globally unique namespaces -- Add context to path -- Use short names -- Building packages -- Installing a package -- Package visibility -- Package member visibility -- Importing package -- Specifying package identifiers -- The dot identifier -- The blank identifier -- Package initialization -- Creating programs -- Accessing program arguments -- Building and installing programs -- Remote packages. Summary -- Chapter 7: Composite Types -- The array type -- Array initialization -- Declaring named array types -- Using arrays -- Array length and capacity -- Array traversal -- Array as parameters -- The slice type -- Slice initialization -- Slice representation -- Slicing -- Slicing a slice -- Slicing an array -- Slice expressions with capacity -- Making a slice -- Using slices -- Slices as parameters -- Length and capacity -- Appending to slices -- Copying slices -- Strings as slices -- The map type -- Map initialization -- Making Maps -- Using maps -- Map traversal -- Map functions -- Maps as parameters -- The struct type -- Accessing struct fields -- Struct initialization -- Declaring named struct types -- The anonymous field -- Promoted fields -- Structs as parameters -- Field tags -- Summary -- Chapter 8: Methods, Interfaces, and Objects -- Go methods -- Value and pointer receivers -- Objects in Go -- The struct as object -- Object composition -- Field and method promotion -- The constructor function -- The interface type -- Implementing an interface -- Subtyping with Go interfaces -- Implementing multiple interfaces -- Interface embedding -- The empty interface type -- Type assertion -- Summary -- Chapter 9: Concurrency -- Goroutines -- The go statement -- Goroutine scheduling -- Channels -- The Channel type -- The send and receive operations -- Unbuffered channel -- Buffered channel -- Unidirectional channels -- Channel length and capacity -- Closing a channel -- Writing concurrent programs -- Synchronization -- Streaming data -- Using for…range to receive data -- Generator functions -- Selecting

from multiple channels -- Channel timeout -- The sync package -- Synchronizing with mutex locks -- Synchronizing access to composite values -- Concurrency barriers with sync.WaitGroup -- Detecting race conditions -- Parallelism in Go -- Summary.
Chapter 10: Data IO in Go -- IO with readers and writers -- The io.Reader interface -- Chaining readers -- The io.Writer interface -- Working with the io package -- Working with files -- Creating and opening files -- Function os.OpenFile -- Files writing and reading -- Standard input, output, and error -- Formatted IO with fmt -- Printing to io.Writer interfaces -- Printing to standard output -- Reading from io.Reader -- Reading from standard input -- Buffered IO -- [Buffered writers and readers] -- Buffered writers and readers -- Scanning the buffer -- In-memory IO -- Encoding and decoding data -- Binary encoding with gob -- Encoding data as JSON -- Controlling JSON mapping with struct tags -- Custom encoding and decoding -- Summary -- Chapter 11: Writing Networked Services -- The net package -- Addressing -- The net.Conn Type -- Dialing a connection -- Listening for incoming connections -- Accepting client connections -- A TCP API server -- Connecting to the TCP server with telnet -- Connecting to the TCP server with Go -- The HTTP package -- The http.Client type -- Configuring the client -- Handling client requests and responses -- A simple HTTP server -- The default server -- Routing requests with http.ServeMux -- The default ServeMux -- A JSON API server -- Testing the API server with cURL -- An API server client in Go -- A JavaScript API server client -- Summary -- Chapter 12: Code Testing -- The Go test tool -- Test file names -- Test organization -- Writing Go tests -- The test functions -- Running the tests -- Filtering executed tests -- Test logging -- Reporting failure -- Skipping tests -- Table-driven tests -- HTTP testing -- Testing HTTP server code -- Testing HTTP client code -- Test coverage -- The cover tool -- Code benchmark -- Running the benchmark -- Skipping test functions -- The benchmark report -- Adjusting N.
Comparative benchmarks -- Summary -- Index.

Sommario/riassunto

An insightful guide to learning the Go programming language About This Book Insightful coverage of Go programming syntax, constructs, and idioms to help you understand Go code effectively Push your Go skills, with topics such as, data types, channels, concurrency, object-oriented Go, testing, and network programming Each chapter provides working code samples that are designed to help reader quickly understand respective topic Who This Book Is For If you have prior exposure to programming and are interested in learning the Go programming language, this book is designed for you. It will quickly run you through the basics of programming to let you exploit a number of features offered by Go programming language. What You Will Learn Install and configure the Go development environment to quickly get started with your first program. Use the basic elements of the language including source code structure, variables, constants, and control flow primitives to quickly get started with Go Gain practical insight into the use of Go's type system including basic and composite types such as maps, slices, and structs. Use interface types and techniques such as embedding to create idiomatic object-oriented programs in Go. Develop effective functions that are encapsulated in well-organized package structures with support for error handling and panic recovery. Implement goroutine, channels, and other concurrency primitives to write highly-concurrent and safe Go code Write tested and benchmarked code using Go’s built test tools Access OS resources by calling C libraries and interact with program environment at runtime In Detail The Go programming language has firmly established itself as a

| | |
|---|---|
| | favorite for building complex and scalable system applications. Go offers a direct and practical approach to programming that let programmers write correct and predictable code using concurrency idioms and a full-featured standard library. This is a step-by-step, practical guide full of real world examples to help you get started with Go in no time at all. We start off by understanding the fundamentals of Go, followed by a detailed description of the Go data types, program structures and Maps. After this, you learn how to use Go concurrency idioms to avoid pitfalls and create programs that are exact in expected behavior. Next, you will be familiarized with the tools and libraries that are available in Go for writing and exercising tests, benchmarking, and code coverage. Finally,... |

2.

| | |
|---|---|
| Record Nr. | UNINA9910795256603321 |
| Autore | D'Anna Delio |
| Titolo | The go workshop : a new, interactive approach to learning go / / Delio D'Anna [and five others] |
| Pubbl/distr/stampa | Birmingham, England ; ; Mumbai : , : Packt, , [2019]<br>©2020 |
| ISBN | 1-83864-015-0 |
| Edizione | [1st edition] |
| Descrizione fisica | 1 online resource (821 pages) |
| Disciplina | 005.133 |
| Soggetti | Go (Computer program language) |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | Includes index. |
| Sommario/riassunto | Cut through the noise and get real results with a step-by-step approach to learning Go programming Key Features Ideal for the Go beginner who is getting started for the first time A step-by-step Go tutorial with exercises and activities that help build key skills Structured to let you progress at your own pace, on your own terms Use your physical print copy to redeem free access to the online interactive edition Book Description You already know you want to learn Go, and the smart way to learn anything is to learn by doing. The Go Workshop |

focuses on building up your practical skills so that you can develop high-performing concurrent applications, or even create Go scripts to automate repetitive daily tasks. You'll learn from real examples that lead to real results. Throughout The Go Workshop, you'll take an engaging step-by-step approach to understanding Go. You won't have to sit through any unnecessary theory. If you're short on time you can jump into a single exercise each day, or you can spend an entire weekend learning how to test and secure your Go applications. It's your choice. Learning on your terms, you'll build up and reinforce key skills in a way that feels rewarding. Every physical print copy of The Go Workshop unlocks access to the interactive edition. With videos detailing all exercises and activities, you'll always have a guided solution. You can also benchmark yourself against assessments, track your progress, and receive content updates. You'll even earn secure credentials that you can share and verify online upon completion. It's a premium learning experience that's included with your printed copy. To redeem it, follow the instructions located at the start of your Go book. Fast-paced and direct, The Go Workshop is the ideal companion for Go beginners. You'll build and iterate on your code like a software developer, learning along the way. This process means that you'll find that your new skills stick, embedded as best practice. A solid foundation for the years ahead. What you will learn Get to grips with Go fundamentals and best practices Learn how to effectively organize your code for production environments Explore how third-party packages can help make you a better engineer Keep your development process bug-free with unit tests and benchmarks Who this book is for Our goal at Packt is to help you be successful, in whatever it is you choose to do. The Go Workshop is an ideal Go tutorial for the Go beginner who is just getti...