



Beginning Go Programming

Build Reliable and Efficient
Applications with Go

—

Rumeel Hussain
Maryam Zulfiqar

Apress®

Beginning Go Programming

**Build Reliable and Efficient
Applications with Go**

**Rumeel Hussain
Maryam Zulfiqar**

Apress®

Beginning Go Programming: Build Reliable and Efficient Applications with Go

Rumeel Hussain
Dubai, United Arab Emirates

Maryam Zulfiqar
Lahore, Pakistan

ISBN-13 (pbk): 978-1-4842-8857-3
<https://doi.org/10.1007/978-1-4842-8858-0>

ISBN-13 (electronic): 978-1-4842-8858-0

Copyright © 2022 by Rumeel Hussain and Maryam Zulfiqar

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Celestin Suresh John
Development Editor: James Markham
Coordinating Editor: Mark Powers
Copy Editor: Kezia Endsley

Cover designed by eStudioCalamar

Cover image by Benjamin Davies on Unsplash (www.unsplash.com)

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub (<https://github.com/Apress>). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*This book is dedicated to Rumeel's late mother
and Abdullah Hassan.*

Table of Contents

- About the Authors.....xi**
- About the Technical Reviewerxiii**
- Prefacexv**

- Chapter 1: Introduction..... 1**
 - Is GoLang Static-Typed or Compiled? 1
 - Compiled Programming Language2
 - Statically-Typed Language2
 - Is Go an Object-Oriented Programming Language?..... 3
 - Features that Make GoLang the Premium Choice for Programming..... 3
 - Intentionally Excluded Features from GoLang4
 - Go Programs5
 - Summary.....5

- Chapter 2: Go Basics 7**
 - Ancestors of Go..... 7
 - Go Syntax..... 7
 - Installing Go 11
 - Go Playground..... 12
 - Developing Go Applications Using IDEs..... 13
 - Getting Started Programming Go Applications..... 14
 - Let’s Print Hello World! 14
 - How to Execute a Go Program 16
 - Keywords..... 17

TABLE OF CONTENTS

Variables	18
Variable Data Types	18
Variable Naming Conventions	20
Declaring Variables	20
Taking User Input	22
Using scanf	23
Using Scanln	24
Using bufio	25
Math Operators and Packages	26
The Math Package	31
Dates and Times	33
Operators Precedence in Go	35
Memory Management and Reference Values	36
New vs Make	37
Incorrect Memory Allocation Example	37
Correct Memory Allocation Example	38
Memory Deallocation	39
Pointers Data Type	39
What Is a Pointer?	39
Declaring Pointers in Go	40
Comparison with Java and C-Style Languages	43
Ordered Values in Arrays and Slices	43
Arrays in Go	44
Slices in Go	48
Maps	57
Defining Maps	57
Adding Entries to a Map Object	57

Deleting Entries from a Map Object.....	58
Iterating Over Stored Values in a Map Object	58
Structs Data Type	62
Defining a Structure	63
Accessing Members of a Structure	63
Passing Structures as Function Arguments.....	65
Pointers to Structures.....	67
Program Flow.....	70
If Statement.....	70
Switch Statement.....	72
For Statement.....	75
The goto Statement.....	78
Functions	79
Defining a Function	80
Doing Function Calls in Go.....	82
Return More than One Value from Functions.....	84
Passing Arguments to Functions	85
Methods.....	88
Write/Read Text Files	91
Write Text Files	91
Read Text Files.....	93
HTTP Package.....	95
JSON	97
Summary.....	102

TABLE OF CONTENTS

Chapter 3: Go Recipes: Programming Fundamentals and Basics.....103

- Numbers and Slices in Go..... 103
- Working with Maps in Go 106
- Go’s Catch of Error Handling 107
- Defer and Panic Recovery..... 110
- Hands-on Challenge..... 114
 - Solution 114
- Summary..... 116

Chapter 4: Working with Text117

- Go String Formatting and Working with Unicode 117
- Case-Insensitive Comparisons in Go..... 119
- Regular Expressions and Reading Text Files with Go..... 121
- Hands-on Challenge..... 129
 - Solution 130
- Summary..... 132

Chapter 5: Structs, Methods, and Interfaces133

- Go Structs, Methods, and Interfaces 133
 - Structs 134
 - Methods..... 140
 - Interfaces 142
- Empty Interface and Working with IOTA in Go..... 148
 - JSON Encoding/Decoding..... 148
 - Generics 151
- Hands-on Challenge..... 153
 - Solution 153
- Summary..... 158

Chapter 6: Working with JSON	159
JSON Package.....	159
Unmarshalling JSON with GO.....	160
Parsing Complex JSON with Go	163
Marshalling JSON with Go	167
Handling Missing Values and Zeros in JSON.....	169
Using mapstructure to Handle Arbitrary JSON.....	172
Summary.....	176
Chapter 7: HTTP	177
Go and HTTP Calls.....	177
Authentication and Writing an HTTP Server in Go.....	182
REST with gorilla/mux.....	187
Hands-on Challenge.....	191
Solution.....	192
Summary.....	203
Chapter 8: Concurrency	205
Understanding Goroutines	206
Converting Sequential Code into Concurrent Code.....	207
Using Goroutines with Shared Resources.....	217
Seeing the Impact of Shared Resources on Goroutines	217
Accessing Shared Resources Using Mutual Exclusion	220
Modifying Shared Resources Using Atomic Counters.....	223
Synchronizing Goroutines	225
sync.WaitGroup and sync.Once	226
Timeouts in Go.....	230
Pooling Goroutines	233

TABLE OF CONTENTS

Hands-On Challenge 236
 Solution 236
 Summary..... 239
Chapter 9: Tricks and Handy Tips241
 Importing Packages 241
 Check What Packages Are Being Imported by Your Application..... 242
 Use goimports Instead of gofmt..... 243
 Code Organization..... 243
 To Use or Not to Use: Custom Constructors 243
 Modularizing and Organizing Code into Packages..... 244
 Dependency Package Management..... 245
 Compiler Optimizations..... 245
 Using Git's SHA to Set the Build ID..... 248
 The Case of Elegant Constants, aka IOTA..... 248
 Auto Increment 250
 Custom Types 250
 Skipping Values 253
 Expressions 253
 Tricky Constants 254
 Summary..... 255
Index.....257