**Ultimate Go Programming:**

**About:**

Ultimate Go Programming LiveLessons, Second Edition, provides an intensive, comprehensive, and idiomatic view of the Go programming language. This course focuses on both the specification and implementation of the language, including topics ranging from language syntax, design, and guidelines to concurrency, testing, and profiling. This class is perfect for anyone who wants a jump-start in learning Go or wants a more thorough understanding of the language and its internals.

**Link to course:**[**Ultimate Go Programming, Second Edition**](https://learning.oreilly.com/videos/ultimate-go-programming/9780135261651/)

**Link to course github repository:**[**gotraining**](https://github.com/ardanlabs/gotraining)

**Links to materials shown in course:**

1. [Lesson 1 - Design Guidelines](https://github.com/ardanlabs/gotraining/tree/master/topics/go)
2. Lesson 2 - Language Syntax
   1. [Variables](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/variables)
   2. [Struct Types](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/struct_types)
   3. [Pointers](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/pointers)
   4. [Constants](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/constants)
3. Lesson 3 - Data Structures
   1. [Data Oriented Design](https://github.com/ardanlabs/gotraining/tree/master/topics/go#data-oriented-design)
   2. [Arrays](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/arrays)
   3. [Slices](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/slices)
   4. [Maps](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/maps)
4. Lesson 4 - Decoupling
   1. [Methods](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/methods)
   2. [Interfaces](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/interfaces)
   3. [Embedding](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/embedding)
   4. [Exporting](https://github.com/ardanlabs/gotraining/tree/master/topics/go/language/exporting)
5. Lesson 5 - Composition
   1. [Grouping Types](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/composition#grouping-types)
   2. [Decoupling](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/composition#decoupling)
   3. [Conversion and Asserions](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/composition#conversion-and-assertions)
   4. [Interface Pollution](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/composition#interface-pollution)
   5. [Mocking](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/composition#mocking)
   6. [Design Guidelines](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/composition#interface-and-composition-design)
6. Lesson 6 - Error Handling

This chapter has a single readme file that goes through different concepts and materials. You can access the full readme [here](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/error_handling). The links in the content will pe to the go playground examples which will be used to talk through this subject.

* 1. [Default Error Values](https://play.golang.org/p/beGEdO2QE4g)
  2. [Error Variables](https://play.golang.org/p/JQUJbS20MrE)
  3. [Type as Context](https://play.golang.org/p/BmiblC2Q7MC)
  4. [Behavior as Context](https://play.golang.org/p/sNRSXKtcJKM)
  5. [Find the Bug](https://play.golang.org/p/CBL-ADH-nSv) | [The Reason](https://play.golang.org/p/-f4PPcBGkDU)
  6. [Wrapping Errors](https://play.golang.org/p/Zt1Z5k4HbDG) | [Wrapping errors with stdlib](https://play.golang.org/p/f5bw9G7OLog)
  7. [Exercises](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/error_handling#exercises)

1. Lesson 7 - Packaging

This chapter has a single readme file that goes through different concepts and materials. You can access the full readme [here](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/packaging). The links to the example service used in the presentation is [here](https://github.com/ardanlabs/service).

* 1. [Language Mechanics](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/packaging#language-mechanics)
  2. [Design Guidelines](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/packaging#design-philosophy)
  3. [Package-Oriented Design](https://github.com/ardanlabs/gotraining/tree/master/topics/go/design/packaging#project-structure)

Extended link to project structures [here](https://github.com/golang-standards/project-layout#go-directories) Since go **1.12** a new way of using 3rd party dependencies has been introduced in go, called go.mod. The official post annoucing it is [here](https://blog.golang.org/using-go-modules) and documentation related to go modules is [here](https://golang.org/ref/mod)

1. Lesson 8 - Goroutines
   1. [OS Scheduler Mechanics](https://www.ardanlabs.com/blog/2018/08/scheduling-in-go-part1.html)
   2. [Go Scheduler Mechanics](https://www.ardanlabs.com/blog/2018/08/scheduling-in-go-part2.html)
   3. [Creating Goroutines](https://github.com/ardanlabs/gotraining/tree/master/topics/go/concurrency/goroutines)
2. Lesson 9 - Data Races

Data races are covered in this [README.md](https://github.com/ardanlabs/gotraining/tree/master/topics/go/concurrency/data_race)

* 1. [Cache Coherency and False Sharing](https://github.com/ardanlabs/gotraining/tree/master/topics/go/concurrency/data_race#cache-coherency-and-false-sharing)
  2. [Synchronization with Atomic Functions](https://play.golang.org/p/5ZtLaX7zxt7)
  3. [Sunchronization with Mutexes](https://play.golang.org/p/-iXzElPBnDM) | [Read/Write Mutex](https://play.golang.org/p/-iXzElPBnDM)
  4. [Race Detection](https://github.com/ardanlabs/gotraining/blob/master/topics/go/concurrency/data_race/example1/example1.go)
  5. [Map Data Race](https://play.golang.org/p/ktWRjcJWNjw)
  6. [Interface Based Race Condition](https://play.golang.org/p/ktWRjcJWNjw)

1. Lesson 10 - Channels

[Link to README](https://github.com/ardanlabs/gotraining/tree/master/topics/go/concurrency/channels)

* 1. [Basic Patterns](https://github.com/ardanlabs/gotraining/blob/master/topics/go/concurrency/channels/example1/example1.go)
  2. [Pooling Pattern](https://github.com/ardanlabs/gotraining/blob/8aa35d4bcb8cd94f7e2f45942f527f0562aaa55f/topics/go/concurrency/channels/example1/example1.go#L111)
  3. [Fan Out Pattern](https://play.golang.org/p/zxzHAHIr3Xj)
  4. [Cancelation Pattern](https://github.com/ardanlabs/gotraining/blob/8aa35d4bcb8cd94f7e2f45942f527f0562aaa55f/topics/go/concurrency/channels/example1/example1.go#L238)

1. Lesson 11 - Concurrency Patterns

[Link to README](https://github.com/ardanlabs/gotraining/tree/master/topics/go/concurrency/patterns)

* 1. [Context](https://github.com/ardanlabs/gotraining/tree/master/topics/go/packages/context)
  2. Failure Detection

1. Lesson 12 - Testing

[Link to README](https://github.com/ardanlabs/gotraining/tree/master/topics/go/testing/tests)

* 1. [Basic Unit Testing](https://play.golang.org/p/F7kXmSfr7AE)
  2. [Table Unit Testing](https://play.golang.org/p/1a2u8omEqrX)
  3. [Mocking Web Server Response](https://play.golang.org/p/SILnu117hak)
  4. [Testing Internal Endpoints](https://play.golang.org/p/CSK7SZEeWf3)
  5. [Example Tests](https://play.golang.org/p/rE0DRliZH9t)
  6. [Sub Tests](https://play.golang.org/p/7PrkFU-qVdY)
  7. [Code Coverage](https://github.com/ardanlabs/gotraining/tree/master/topics/go/testing/tests#coverage)

1. Lesson 13 - Benchmarking

[Link to README](https://github.com/ardanlabs/gotraining/tree/master/topics/go/testing/benchmarks)

* 1. [Basic Benchmarking](https://github.com/ardanlabs/gotraining/blob/master/topics/go/testing/benchmarks/basic/basic_test.go)
  2. [Sub Benchmarking](https://github.com/ardanlabs/gotraining/blob/master/topics/go/testing/benchmarks/sub/sub_test.go)
  3. [Validate Benchmarking](https://github.com/ardanlabs/gotraining/blob/master/topics/go/testing/benchmarks/validate/validate_test.go)

1. Lesson 14 - Profiling and Tracing

[Link to README](https://github.com/ardanlabs/gotraining/tree/master/topics/go/profiling)

* 1. Profiling Guidelines
  2. Stack Traces
  3. Micro Level Optimization
  4. Macro Level Optimization
  5. Execution Tracing