

# Welcome Gophers!

## What You Need to Know

- Questions during this workshop should be directed to the Teams chat or the Discord channel associated with your instructor.
- Please ask for permission before sharing your screen.
  - If you are on a Mac, sharing your screen may only be possible if you update your Mac's security and privacy settings. Go to the Apple menu > System Preferences, click Security & Privacy, then click Privacy and select Microsoft Teams.
- Breaks will be given near the top of every hour. During the break, a timer will be displayed so you are aware when the session will continue.
- If you are experiencing streaming issues, exit Teams, restart your computer, and then re-enter the workshop. In the case quality issues persist, please take note of the TA's email address located in the bottom right of this screen.
- This workshop is being recorded! If you don't want to be recorded for privacy reasons, don't share your camera and microphone.



**Luke McCoy**  
Google

**Alex Mammay, TA**  
alexmammay@gmail.com

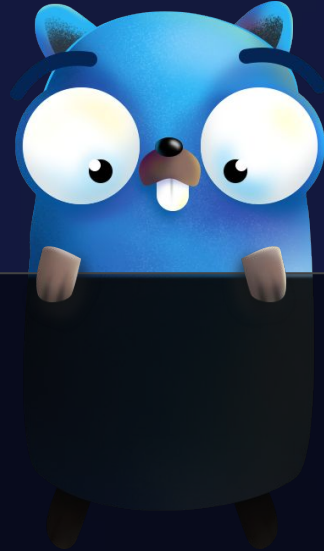
**Getting a Jumpstart in Go**

**GopherCon**

# Code blocks example

```
package main
import(
    "fmt"
)

func main(){
    fmt.Println("This is Cool")
}
```





# Go Jumpstart

Google Cloud

# Synopsis

Want to get a jumpstart learning Go? Maybe you are a systems integrator that is more of a scriptor than a developer? Or maybe you'd like to take a fresh look at Go from the beginning. The goal of this workshop is to give you a good foundation on which to build your Go development skills. We will cover the fundamentals of Go to help you boost your programming productivity out of the gate. The two-day course seeks to teach the language by reviewing and discussing source code line by line.

# Schedule Day One

<b>Setup Go Background</b> (12 to 12:45)	Workstation setup (Go, IDE), your first app, language overview and exercises
<b>Hello World</b> (1 - 1:45)	Basic structure of a .go file. Run and compile a basic program
<b>Roadwork</b> (2 -2:45)	Look at how some basic things are constructed and create an http service
<b>Roadwork</b> (3 -3:45)	Let's write some code together
<b>Bonus and Overflow</b> 4-5	We will be available to answer questions, have general discussions and plan tomorrow's sessions



# Workstation setup

Google Cloud

# Setup Go



Install Go for your platform from <https://golang.org/dl/>



Google Cloud

# IDE Install Your Favorite IDE

- **Atom** - <https://atom.io>
- **Visual Studio Code** - <https://code.visualstudio.com/> (Recommended for Beginners)
- **Goland** - <https://www.jetbrains.com/go/specials/go/go.html?dclid=CJzE5LDG4NwCFevuwQodWy4Pow>
- **LiteIde** - <https://github.com/visualfc/liteide>
- **VIM** - <https://github.com/fatih/vim-go>
- **Emacs** - <https://github.com/dominikh/go-mode.el>



# Explore Go commands

## The commands are:

**bug** start a bug report

**build** compile packages and dependencies

**clean** remove object files and cached files

**doc** show documentation for package or symbol

**env** print Go environment information

**fix** update packages to use new APIs

**fmt** gofmt (reformat) package sources

**generate** generate Go files by processing source

**get** download and install packages and dependencies

**install** compile and install packages and dependencies

**list** list packages or modules

**mod** module maintenance

**run** compile and run Go program

**test** test packages

**tool** run specified go tool

**version** print Go version

**vet** report likely mistakes in packages

# Getting the samples code and slides

## Commands

- `cd <<clone location>>`
- `git clone https://github.com/goog-lukemc/gotrain`
- Slide and docs are in gotrain/assets
- Source is in gotrain/<various folders>



# Background

Google Cloud

# O2

# History

## Who

Robert Griesemer  
Rob Pike  
Ken Thompson

## Why

Combine the ease of a dynamic type language with the safety of the static type system.  
[https://golang.org/doc/faq#Is\\_Go\\_an\\_object-oriented\\_language](https://golang.org/doc/faq#Is_Go_an_object-oriented_language)  
[https://golang.org/doc/effective\\_go.html](https://golang.org/doc/effective_go.html)  
<https://golang.org/doc/code.html>

## Where are we now

1.15.4 Nov 2020  
<https://golang.org/dl/>

## More info

<https://talks.golang.org/2012/splash.article>  
<https://tip.golang.org/doc/go1.11>  
<https://golang.org/doc/devel/release.html>  
<https://talks.golang.org/2015/gophercon-goevolution.slide#8>  
<https://golang.org/doc/faq>

# What's cool about Go (top 3)

## ► Concurrency: (tomorrow)

- Concurrency is not parallelism ([https://www.youtube.com/watch?v=cN\\_DpYBzKso](https://www.youtube.com/watch?v=cN_DpYBzKso))
- Concurrency is about having the best design to maximize parallelism if it is available.

## ► Interfaces: (tomorrow)

- We will do an overview of design and implementation Monday

## ► Portability: (today)

- Go is not runtime interpreted (There is nothing to install on the target to execute a Go program.)
- A simple build switch can build the executable for any supported platform.

# Go Is Good for?

## Coders

Writing code to solve the problem in front of you. Slinging code - having fun!

## Developers

Writing code to solve a problem for generic reuse. Writing small - having fun!

## Idiomatic

What is this anyway?

[https://golang.org/doc/effective\\_go.html](https://golang.org/doc/effective_go.html), <https://golang.org/doc/code.html>,  
<https://github.com/golang/go/wiki/CodeReviewComments>

# Schedule Day Two

<b>Testing</b> (12 to 12:45)	Table Driven and other testing
<b>Interfaces</b> (1 - 1:45)	Creating and using and interface
<b>Interfaces</b> (2 - 2:30)	Review a code for advanced use of an interface
<b>Concurrency</b> (2:45 - 3:45)	Concurrency Flyover
<b>Bonus and Overflow</b> 4-5	We will be available to answer questions, have general discussions and plan tomorrow's sessions



Google Cloud

# Yesterday's Follow Ups

## Pointers

<https://medium.com/@vCabbage/go-are-pointers-a-performance-optimization-a95840d3ef85>

## More Exercises

We have added more complex exercises for today





# Hello, World

Google Cloud



# ~/gotrain/hello

**Review:** main.go (In Editor)

**Build:** main.go (In Editor)

**Build for any platform from any platform:**

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

**Running the file:** `go run main.go`

# ~/gotrain/hello\_flag

**Review:** main.go (In Editor)

**Build:** main.go (In Editor)

**Build for any platform from any platform:**

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

**Running the file and flow testing:** `go run main.go`

# ~/gotrain/hello\_struct

**Review:** main.go (In Editor)

**Build:** main.go (In Editor)

## Build for any platform from any platform:

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

## Running the file and flow testing:

- `go build`
- `go run main.go`



# The basics

Google Cloud

# 04

# errors ~/gotrain/errors

**Review:** main.go (In Editor)

**Build for any platform from any platform:**

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

**Running the file and flow testing:** `go run main.go`

# basic ~/gotrain/basics

**Review:** main.go (In Editor)

**Build for any platform from any platform:**

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

**Running the file and flow testing:** `go run main.go`

# basic ~/gotrain/asciicoolness

**Review:** main.go (In Editor)

**Build for any platform from any platform:**

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

**Running the file and flow testing:** `go run main.go`



# basic ~/gotrain/basichttpserver

**Review:** main.go (In Editor)

**Build for any platform from any platform:**

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

**Running the file and flow testing:** `go run main.go`

# Practical Work

Go Cloud Libs:

<https://github.com/googleapis/google-cloud-go>

How to submit bugs and contribute

<https://github.com/golang/go/blob/master/CONTRIBUTING.md>

Language Spec

<https://golang.org/ref/spec>

Effective Go

[https://golang.org/doc/effective\\_go.html](https://golang.org/doc/effective_go.html)

When to Panic

<https://eli.thegreenplace.net/2018/on-the-uses-and-misuses-of-panics-in-go/>



Google Cloud



# Testing

Google Cloud

# 05

# test ~/gotrain/onetest

**Review:** main.go (In Editor) and main\_test.go (In Editor)

## Build for any platform from any platform:

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

## Running the tests:

- `go test`

# test ~/gotrain/twotest

**Review:** main.go (In Editor) and main\_test.go (In Editor)

## Build for any platform from any platform:

- **Pi:** env GOOS=linux GOARCH=arm go build -v main.go -o program-arm
- **MAC:** env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64
- **Windows:** env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe

## Running the tests:

- go test

# test ~/gotrain/awesomeexample

**Review:** awesome.go (In Editor) and example\_awesome\_test.go (In Editor)

## Build for any platform from any platform:

- **Pi:** env GOOS=linux GOARCH=arm go build -v main.go -o program-arm
- **MAC:** env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64
- **Windows:** env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe

## Running the test and doc server:

- go test
- godoc -http=:8080



# Interfaces

Google Cloud



# Interfaces

## Web links

- <https://gobyexample.com/interfaces>
- <https://medium.com/golangspec/interfaces-in-go-part-i-4ae53a97479c>

**Let's do some code review**





# Concurrency

Google Cloud

# 07

# concurrency ~/gotrain/basichttpserver

**Review:** main.go (In Editor)

**Build for any platform from any platform:**

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

**Running the file and flow testing:** `go run main.go`

# concurrency ~/gotrain/sametime ~/gotrain/mutex ~/gotrain/sametime

**Review:** main.go (In Editor)

**Build for any platform from any platform:**

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

**Running the file and flow testing:** `go run main.go`



# Testing

Google Cloud



Let's make  
something  
TDD style

Survey: <https://goo.gl/forms/tj22lUtiqEODfHP2>

**We done**