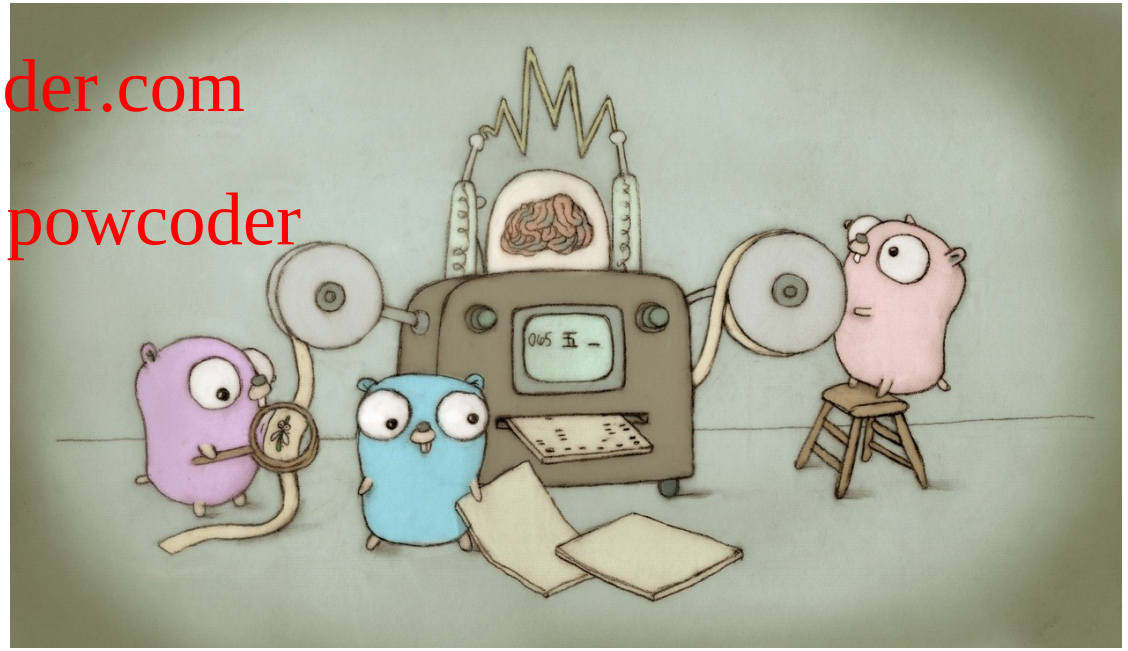
A popular procedural language with support for objects and concurrency.

2009

CMPT 383

Summer 2022

SFU Surrey



Artwork by Renee French

What was going on in 2009?

2009: First Appearance

Bitcoin

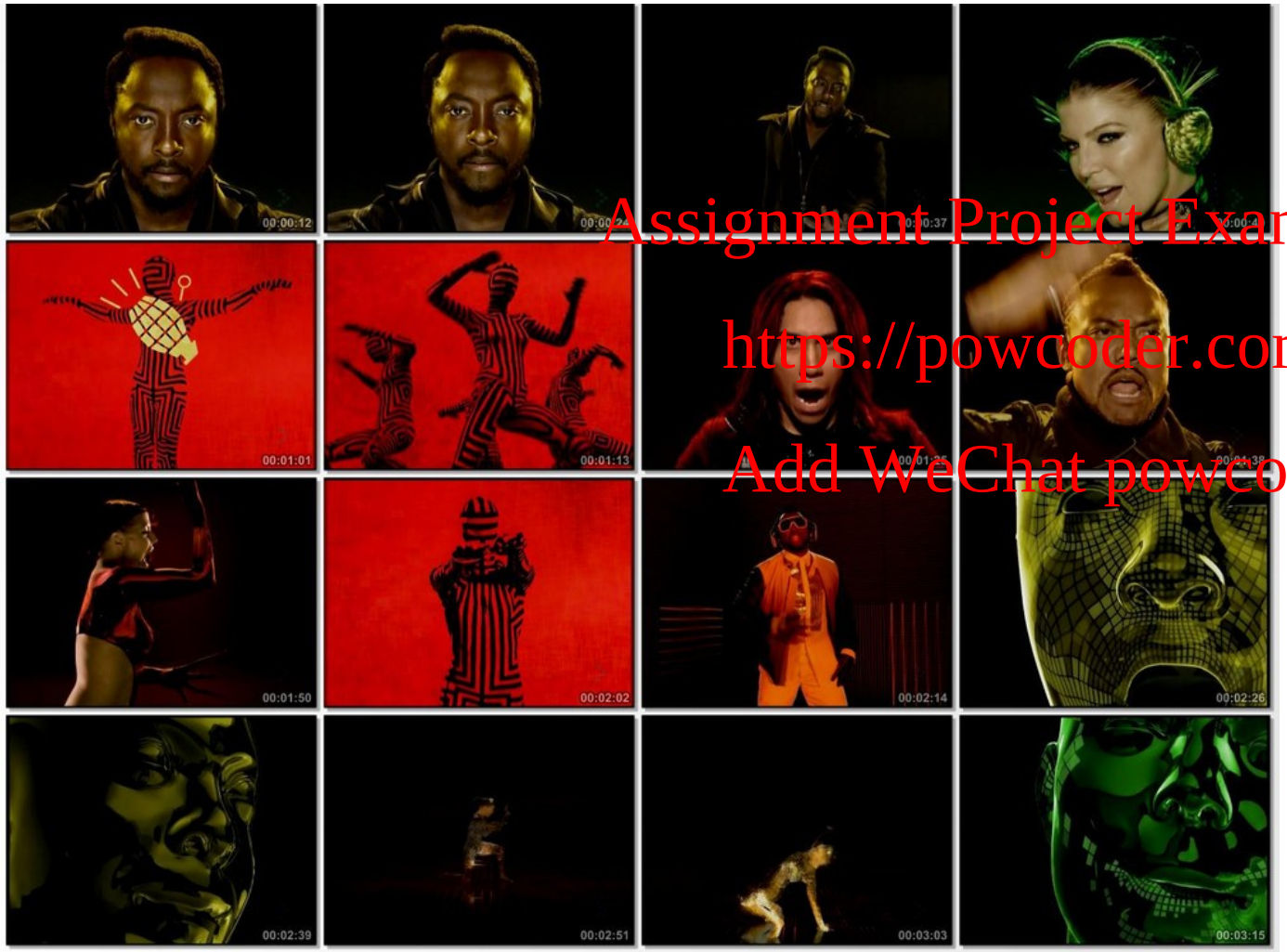
2009: Most Popular Console Video Game

2009: Highest-Grossing Movie

Transformers

Revenge of the Fallen

2009: Most Popular Song



Boom Boom Pow,
The Black Eyed Peas

2009 Facts

- 13th of February, 2009 at 11:31:30pm: Unix time passed 1,234,567,890 seconds
- H1N1 flu declared a global pandemic
- Usain Bolt set 100m world record of 9.58s
- Popstar Michael Jackson died.
- On June 28, 2009 physicist Stephen Hawking threw a party for time-travelers. He announced the party the day after it happened, and according to Hawking, no one came.

Go

Created at Google in 2007

First public release 2009

Hey! Ho! Let's Go!

Tuesday, November 10, 2009

Here at Google, we believe programming should be fast, productive, and most importantly, fun. That's why we're excited to open source an experimental new language called Go. Go combines the development speed of working in a dynamic language like Python with the performance and safety of a compiled language like C or C++. Typical builds feel instantaneous; even large binaries compile in just a few seconds. And the compiled code runs close to the speed of C. Go lets you move fast.

Go is a great language for systems programming with support for multi-processing, a fresh and lightweight take on object-oriented design, plus some cool features like true closures and reflection.

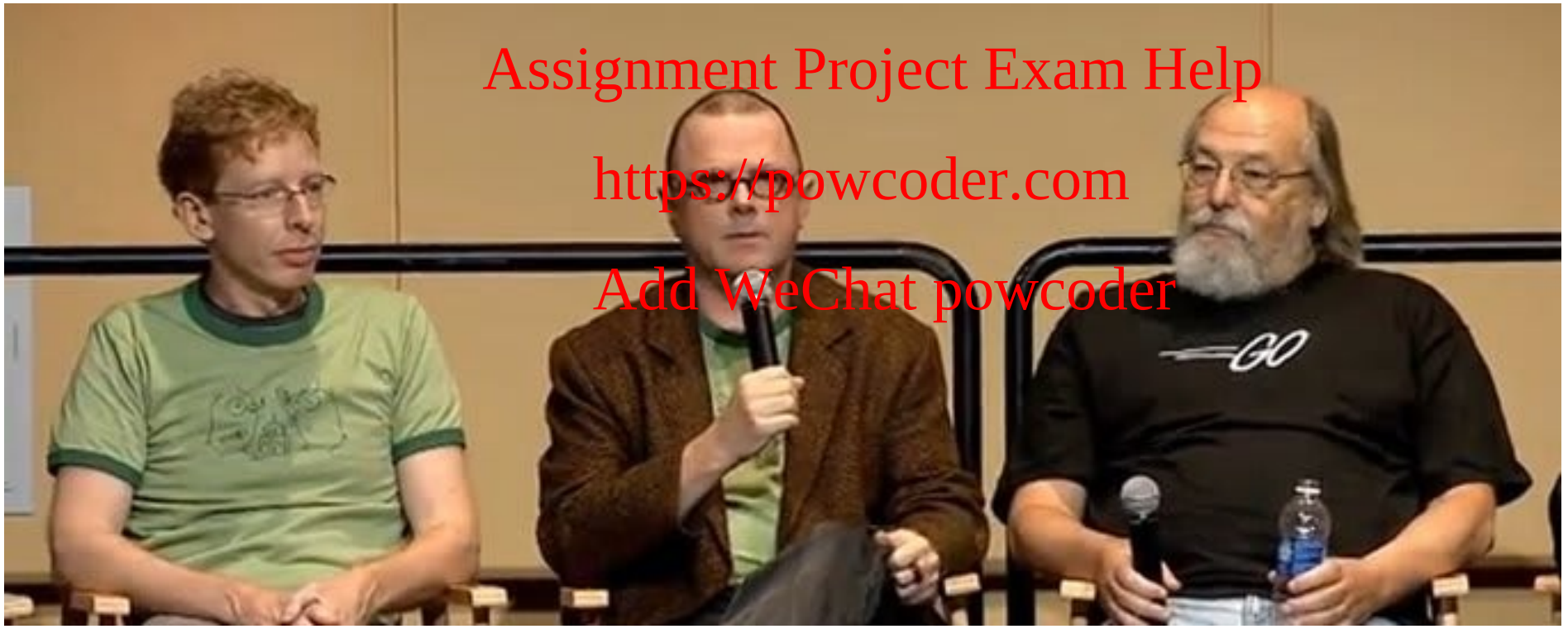
Want to write a server with thousands of communicating threads? Want to spend less time reading blogs while waiting for builds? Feel like whipping up a prototype of your latest idea? Go is the way to go! Check out the video for more information or visit golang.org.

By Robert Griesemer, Rob Pike, Ken Thompson, Ian Taylor, Russ Cox, Jini Kim and Adam Langley - The Go Team

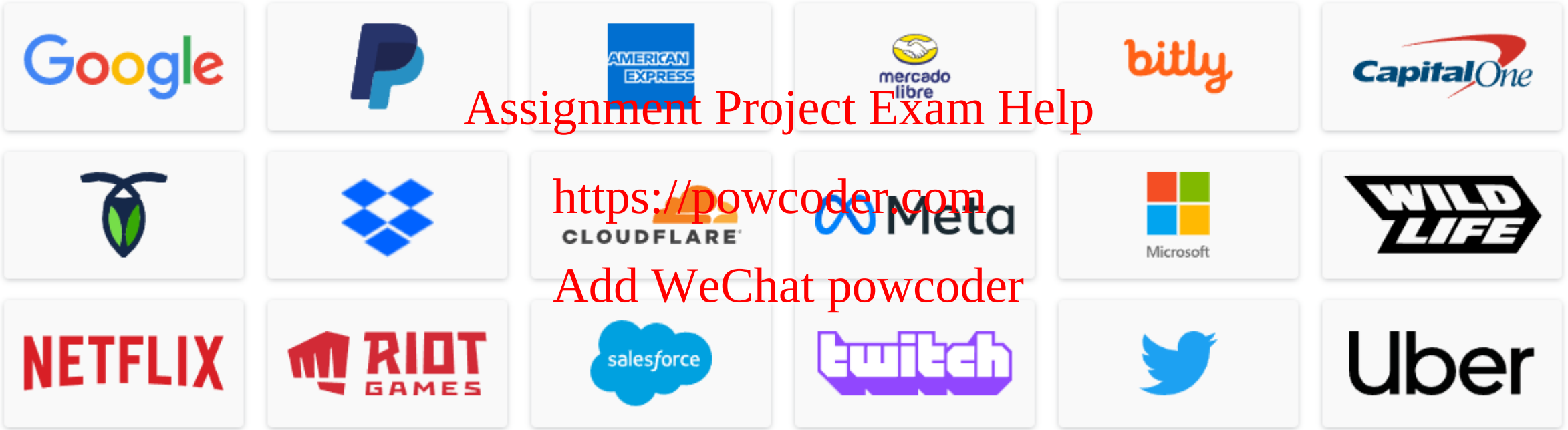
Go’s Original Designers

One of the original designers of Unix. Also won a Turing Award!

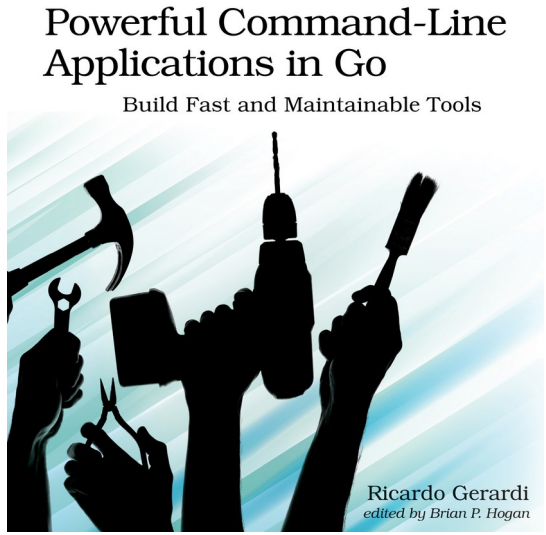
Robert Griesemer Rob Pike Ken Thompson



Go Can Get You a Job!



What's Go Good For?



Command-line utilities.

Websites and servers
e.g. Hugo is a popular
static website generator

docker

kubernetes

DevOps tools
Like Docker and
Kubernetes

What's Go Not so Good For?

- Go is **garbage collected**, so you don't get same flexibility as C/C++ when it comes to memory management
- Go is generally not as efficient as C, C++, Rust, so it is not the best choice when performance is the primary concern
- Go discourages "clever" code in favour of straightforward, simple code that some programmers think is not as "fun"

Measuring the performance of different programming languages is surprisingly difficult to do fairly. Check out the Computer Language Benchmarks Game

.

Go was designed by software engineers needing to solve real-world problems. They weren't programming language theoreticians.

Major Features of Go

Syntactically similar to C, but with some important differences

- Multiple ways to define new variables
- Optional semi-colons; some syntax simplifications

Compiled

- Fast compile times are one of the distinctive features of Go

Apparently the decision to start the Go project was made while waiting for a C++ program to compile.

Memory-safe and garbage-collected

- No dangling pointers, no manual memory deletion

Uses interfaces and embedding instead of inheritance

- Lack of inheritance is not usually a problem, but it can take some getting used to at first

Major Features of Go

Types

- Statically typed: type of values is known at run-time and checked by the compiler
- Basic numeric types
- Pointers
- Arrays
- Slices: “point” to a sequence of elements in an array
- Built-in hash maps
- Structures and interfaces

Functions

- Can return multiple values
- Methods are functions with a special “self” parameter that can be used with interfaces
- Closures: functions can be passed to functions and returned by functions

Packages and Modules

- Organized collections of code
- Support for downloading from online

Closures are a major feature we’ll see again in other languages.

Major Features of Go

Many programmers say Go's good built-in support for concurrency is its killer feature.

Concurrency

- Based the **CSP** model of concurrency
- Goroutines allow you to run multiple functions concurrently
- Channels communicate information between goroutines in a structured way

Extensive Standard Library

- Go is ready to write web servers and apps out of the box

Software tool support

- Syntax of Go was deliberately made as simple as possible to make it easier for software tools to process it
- Go source code has a standard format; use the **gofmt** tool
- Other standard tools, such as support for unit tests, fuzzing

Generics

- Added in Go 1.18 in 2022; we probably won't cover them in this course

Major Features Not in Go

Features **not** part of Go

- No inheritance
- No pointer arithmetic
- No assertions
- No implicit type conversions
- No C++-style exceptions; instead, defer/panic and functions return error values
- No interpreter, compiled only

Go aims to be simple yet practical. The core rules of Go programming are pretty small compared to other modern languages.

Hello, World!

```
$ go run hello_world.go
$ ./hello_world
```

```
// hello_world.go

package main

import "fmt"

func main() {
    fmt.Println("Hello, world!")
}
```