

Go Language Report

Jeremiah Webb

Embry-Riddle Aeronautical University, CS332

Fall 2022

CS332: Report 1: Turing Computable Language

Google developed Go in 2007, which is a multi-paradigm language, however, it is used primarily as a procedural language, however, it can also support object-oriented programming, functional programming, and concurrent programming. Google software engineers wanted to create a dependable, efficient programming language with the simplicity of Python but statically typed, explicit and compiled into machine code, like C.

The Go programming language gets its roots from Ken Thompson, the inventor of the B & C programming languages. Its initial public 1.0 release was in 2012 and is currently maintained by Google. It takes lessons from more famous languages such as Python, C++, and C to ensure ease of use in “memory safety, portability, and quick compilation” (Doxsey, 2016). With the promise of compatibility using newer versions of Go, nearly all programs written in older versions will be able to compile and run. As more and more multicore processors are being developed and used today in consumer products, Go’s unique “goroutine” is capable of “running concurrently with other functions”(Doxsey, 2016). Thus, Go is Google’s answer to the ever demanding programming language competition.

The main communities that use the Go programming language are web developers, cloud software developers, and cybersecurity engineers. These communities benefit from its “automatic garbage collection, compilation and execution speeds, and no virtual machine needed” (Doxsey,2016). Go’s ability to be compiled and ran quickly makes it a top choice within cloud developers, as Amazon Web Services made it a popular choice for it’s Lambda service, where one can run written code inside the “cloud”. Additionally, the language is famously known for its use in container services, such as Docker and Kubernetes. Both technologies are used in cloud development and web development for computer auto balancing and data processing. Furthermore, Go just released in 2022 the use of generics in its functions, which allow for generic code for parameterized types. Additionally, communities enjoy

CS332: Report 1: Turing Computable Language

the “gofmt” tool for static code analysis. Go will not compile if “not using declared variables and importing unused packages”. Likewise, embedded developers can enjoy the use of Go’s pointer system, where developers can access memory manually for embedded applications that can compete with Rust and C/C++.

In contrast, however, there are some valid criticisms of Go. There is a lack of runtime safety and error-handling abilities for a developer. Due to no try, catch, statements, there can be repeated code. In addition, Go has no immutability declarations for variables, developers can accidentally overwrite a value that is to be constant and vice versa. Go is relatively new, and generics were just now released in 2022, this can cause an issue for refactoring of code. Moreover, with the ability to return multiple items in a function, one can return a specific error, however, a developer will have to explicitly check that the error does not exist in the return data and have to handle that error. A developer can simply assume there was no error and then proceed with the program, this can cause unhandled runtime errors.

Syntax Table

Declaration	Unsigned 8-bit integers	uint8
	Unsigned 16-bit integers	uint16
	Unsigned 32-bit integers	uint32
	Unsigned 64-bit integers	uint64
	Signed 8-bit integers	int8
	Signed 16-bit integers	int16
	Signed 32-bit integers	int32
	Signed 64-bit integers	int64
	IEEE-754 32-bit floating-point numbers	float32
	IEEE-754 64-bit floating-point numbers	float64
	Complex numbers with float32 real & imaginary parts	complex64
	Complex numbers with float64 real & imaginary parts	complex128
	String	String

CS332: Report 1: Turing Computable Language

Assignment Selection	->	var name datatype = value
	IF	If condition { code } else if condition { code } else { code }
	Select Case Pattern Match (Switch)	switch i { case x: code case y: code default: code }
Repetition	WHILE (Go does not have a dedicated keyword for While)	for condition { code }
	FOR	For variable; condition; increment{ code }
	Infinite loop	For { }

CS332: Report 1: Turing Computable Language

Works Cited

Doxsey, C. (2016). *Introduction to Programming in Go*. O'Reilly Media.

(ISBN-13: 9781478355823)