

Top 20 GO Programming (Golang) Interview Questions & Answers

1) Explain what is GO?

GO is an open source programming language which makes it easy to build simple, reliable and efficient software. Programs are constructed from packages, whose properties allow efficient management of dependencies.

2) What is syntax like in GO?

Syntax in GO is specified using Extended Backus-Naur Form (EBNF)

- Production = production_name “=” [Expression]
- Expression = Alternative { “|” Alternative }
- Alternative = Term { Term }
- Term = Production_name | token [“...”token] | Group | Option | Repetition
- Group = “ (“ Expression”)”
- Option = “ [“ Expression “]”
- Repetition = “ { “ Expression “ }”

3) Explain what is string literals?

A string literals represents a string constant obtained from concatenating a sequence of characters.

There are two forms,

- Raw string literals: The value of raw string literals are character sequence between back quotes “. The value of a string literal is the string composed of the uninterrupted character between quotes.
- Interpreted string literals: It is represented between double quotes “. The text between the double quotes which may not contain newlines, forms the value of the literal.

4) Explain packages in Go program?

Every GO program is made up of packages. The program starts running in package **main**. This program is using the packages with import paths “**fmt**” and “**math/rand**”.

5) Explain workspace in GO?

Inside a workspace GO code must be kept. A workspace is a directory hierarchy with three directories at its root.

- src contains GO source files organized into packages
- pkg contains package objects and
- bin contains executable commands



6) Explain how to use custom packages in GO language?

If you are making your library a separate go get -table project and if your library is for internal use then you can code like this

- Under the directory of your project place the directory with library files
- Refer to the library using its path relative to the root of your workspace consisting the project

For example,

src/

myproject/

mylib/

mylib.go

...

main.go

Now, in **main.go** you could **import myprojec/mylib**.

7) Explain what is GOPATH environment variable?

The GOPATH environment variable determines the location of the workspace. It is the only environment variable that you have to set when developing Go code.

8) Explain how you can do testing in GO?

It has a lightweight testing framework consists of the **go test** command and the **testing** package.

To write a test you have to create a file with a name ending in `_testing`. Go which contains functions named `TestXXX` with signature `func (t *testing.T)`. The test framework runs each such function.

9) Explain what is string types?

A string type represents the set of string values, and string values are sequence of bytes. Strings once created is not possible to change.

10) What are the advantages of GO?

- GO compiles very quickly
- Go supports concurrency at the language level

- Functions are first class objects in GO
- GO has garbage collection
- Strings and Maps are built into the language

11) List out the built in support in GO?

The available built-in-support in GO includes

- Container: container/list , container/heap
- Web Server: net/http
- Cryptography: Crypto/md5 , crypto/sha1
- Compression: compress/ gzip
- Database: database/sql

12) Explain what is go routine in GO? How you can stop go routine?

A goroutine is a function which is capable of running concurrently with other functions

To stop goroutine, you pass the goroutine a signal channel, that signal channel is used to push a value into when you want the goroutine to stop. The goroutine polls that channel regularly as soon as it detects a signal, it quits.

```
Quit := make(chan bool)
```

```
go func () {
```

```
for {
```

```
select {
```

```
case <- quit:
```

```
return
```

```
default
```

```
// do other stuff
```

```
}
```

```
}
```

```
}()
```

```
// Do stuff
```

```
// Quit goroutine
```

```
Quit <- true
```

13) Explain how you can write multiline strings in GO?

To write multiline string in GO you can use a raw string literal, where the string is delimited by back quotes rather than double quotes.

```
' line 1
```

line 2

line 3 '

14) Explain how you to access command line arguments passed to a GO program?

You can access the command line argument using the `os.Args` variables. For example,

```
Package main
```

```
import (
```

```
    "fmt"
```

```
    "OS"
```

```
)
```

```
func main () {
```

```
    fmt.Println(len(os.Args), os.Args)
```

```
}
```

15) Explain how pointer is represented in GO?

In GO a pointer is represented by using the * (asterisk) character followed by the type of the stored value.

16) How you can format a string without printing?

To format a string without printing you have to use command

```
return fmt.Sprintf ("at %v, %s" , e.When , e.What )
```

17) Explain how arrays in GO works differently then C ?

In GO Array works differently than it works in C

- Arrays are values, assigning one array to another copies all the elements
- If you pass an array to a function, it will receive a copy of the array, not a pointer to it
- The size of an array is part of its type. The types `[10] int` and `[20] int` are distinct

18) Explain GO Interfaces ?

In GO, interfaces is a way to specify the behaviour of an object. An interface is created by using the **"type"** word, followed by a name and the keyword `interface`. An interface is specified as two things.

- A set of methods
- Also it is referred as type

19) Explain what Type assertion is used for and how it does it?

Type conversion is used to convert dissimilar types in GO. A type assertion takes an interface value and retrieve from it a value of the specified explicit type.

20) In GO language how you can check variable type at runtime?

A special type of switch is dedicated in GO to check variable type at runtime, this switch is referred as type switch. Also, you can switch on the type of an interface value with Type Switch.

[Guru99](#) Provides [FREE ONLINE TUTORIAL](#) on Various courses like

Java	MIS	MongoDB	BigData	Cassandra
Web Services	SQLite	JSP	Informatica	Accounting
SAP Training	Python	Excel	ASP Net	HBase
Project Management	Test Management	Business Analyst	Ethical Hacking	PMP
Live Project	SoapUI	Photoshop	Manual Testing	Mobile Testing
Data Warehouse	R Tutorial	Tableau	DevOps	AWS
Jenkins	Agile Testing	RPA	JUnit	Software Engineering
Selenium	CCNA	AngularJS	NodeJS	PLSQL

**Stay updated with new
courses at Guru99
Join our Newsletter**