By OnlineInterviewQuestions.com

Go Programming Interview Questions

Need some help when planning your career? Need to enhance and develop your career in the **Go language**? You are in the correct place. In today's time, there is quite tough competition in the market to crack the interview and get a job in an MNC. **Go programming language** is designed with systems programming in mind. It is strongly and statically typed which provides inbuilt support for the garbage collection and supports concurrent programming.

Here are some of the questions, which are asked in **Go Programming Interview** and which will give you a head begin. You never know you might get these set of questions in your next Go programming interview.

Q1. What are Goroutines?

Goroutines are the methods and functions that run concurrently with other methods or functions. Goroutines can be considered as a lightweight thread. These are multiplexed to a fewer number of OS threads. There might be only one thread in a program with thousands of goroutines.

Q2. What are channels in Go language and how can you use them?

Go provides a mechanism, which is called channel that is utilized to share data between goroutines. When you execute a concurrent activity as a goroutine a resource or data is required to share between goroutines, channel acts as a pipe between the goroutines and gives them a mechanism that guarantees a synchronous exchange. There are two types of channels based on their actions:
1. **Unbuffered Channels:** It is used to behave in a synchronous communication between goroutines.
2. **Buffered Channels:** It is used to perform synchronous communication

Q3. How can you distribute tasks in Go to different machines?

Following details is the way how we can distribute various tasks in Go to different machines:
1. Get the Dependencies
2. Writing tasks and worker
3. Requesting/Sending tasks
4. Make sure server is running

Q4. What is a string literal in Go programming?

Go supports two syntaxes for string literals. With regular quotes, special sequences like newlines are interpreted

as actual newlines. Strings literal deserves special consideration because they are implemented very separately in Go as compared with other languages. Strings literals are placed between double quotes "….". Default encodes strings in Go, and moreover, UTF-8 supports ASC11 character set. So there is no need to be worried about encoding. String literal can improve a program's readability.

Q5. What is the default value of type bool in Go Programming?

No there is no way to specify a default value. This is completed on purpose to enhance readability at the cost of little more time on the writer's end. The proper approach of having a "default" is to have a new function, which provides that default to the more function. By applying this, your code becomes clearer on your intent.

Q6. Explain what GOPATH environment variable is?

The GOPATH environment variable indicates the specific location of your workspace. That is the place you find the Go devices, where you create and where you introduce 3rd part packages and binaries. If no GOPATH is set, it is pretended to be user profile on windows and home on UNIX systems.

Q7. How many Looping constructs are present in Go programming language?

There might be a circumstance when you have to execute a block of code several numbers of times. Programming languages give different control structures that take into account more complicated execution ways.
A loop statement enables us to execute a statement or group of statements various times and following is the general type of a loop statement in the majority of the programming languages.

- **Loop Control Statements:**Loop control statements change execution from its normal sequence. At the point when execution leaves its scope, every programmed object that was made in that scope is diminished.
- **The Infinite Loop:** A loop turns into an infinite loop if its condition never turns out to be false. Loop is generally utilized for this reason. You can make an endless loop by remaining the conditional expression empty.

Q8. Is it recommended to use Global Variables in a program that implements Goroutines?

You should **not use global variables** at all. Unless there is a solid reason for that. And you should not use goroutines just for the sake of it, even though go ones are quite cheap they can have side effects and complicate the program needlessly.

Q9. Explain what a modular programming language is.

Modular programming is a strategy for creating software by isolating the functionality of a program into independent, exchangeable modules that are joined together to have the final working software. Modular programming is an answer to the issue of large projects that are hard to troubleshoot and maintain. Modular programming has a main module and numerous auxiliary modules. The primary module is accumulated as an executable, which calls the auxiliary module functions. Auxiliary modules exist as isolated executable files,

which load when the primary EXE runs. Modular programming idea is good with object-oriented programming (OOP) to a large degree.

Q10. What is slice in Go?

Go Slice is deliberation over Go Array. As Go Array enables you to define the type of factors that can hold various information things of a similar kind yet it doesn't give any inbuilt strategy to build size of it progressively or get a sub-exhibit of its own. Slice covers this constraint. It gives numerous utility functions required on Array and is generally utilized as a part of Go programming.

Q11. How to define a structure in Go programming language?

The structure is one of the user-defined data types which is available in Go programming, which allows you to combine data items of various kinds. The structure is that which you should consider the type and struct statements. The type statement binds a name with the type which is a struct in our case. Struct statement defines a new data type, with multiple members for your program. Once a structure is set, it can be used to declare variables also. You can pass a structure as function arguments in a similar way as you give any other variable or pointer.

Q12. What are Interfaces to GO Language?

In Go language, an interface is a method to identify the behavior of an object. An interface is developed with the help of "type" word followed by the name and the keyword interface. Interface values are used to represent as a two-word pair by providing data about the information, which is stored in the interface and a pointer to the related data.

Q13. What are packages in Go language?

Every GO program is combined up of packages. Packages are utilized to organize GO source code for better reusability and readability. These offers compartmentalization of code, and it becomes easy to maintain go applications. "fmt" is the abbreviation of a package which included a variety of functions which are related o formatting and output to the screen.

Q14. List some of the advantages of GO language?

GO is a general-purpose language, which is designed with systems programming in mind. Various advantages of using GO programming language are as follows-:
- It is compiled language which is known for their speeds, as they are directly converted into the machine –level code that is directly read by a computer.
- It has error checking feature which can detect variables that went unused, missing packages, mistyped or invalid operations.
- Go supports concurrency and interfaces at the language level.
- Simplicity, conciseness, and safety.

Q15. Explain why Type assertion is used In GO Language.

Type assertion is utilized to check the values, which are held by interface type variable either implement the

desired interface or is of a concrete type. Type assertion is used to convert different kinds of GO. A type assertion takes an interface value and retrieves from the value of the specified, explicit type.

Q16. Explain pointers in Go?

A pointer is a kind of variable which stores the memory address of another variable. Various programming tasks are performed quite easily with pointers and various other functions such as call by reference which cannot be achieved without using pointers. Any variable or constant you must declare it to a pointer before you can use it to store any variable address.

Q17. What are the function closures?

A function is a group of statements, which performs a task together. You can divide your code into various functions.GO programming supports unidentified or anonymous functions, which can act as function closures.

Q18. What is L-value and R-value in Go?

"L-value" known as a memory location recognizes an object. L-value may show up as either left hand or right-hand side of an assignment operator (=). L-value frequently represents as an identifier.

"R-Value" represents a data value that is stored at some address in memory. An r-value is an articulation that can't have a value assigned to it which means r-value can show up on the right-hand side but not on the left-hand side of a task operator(=).

Q19. What is type casting in Go Programming?

Typecasting is one, which allows programmers to change an entity from one type to another. Or in another way typecasting is a method to convert a variable from one data type to another one.

Q20. What are maps in Go?

Maps in Go programming is an unordered collection of the keys and its associated values. They are quite useful for looking up values fast. A map is like an array instead of an integer index; you can have string keys or other data types to represent keys.

Q21. What is the default value of a local variable in Go?

A variable is one of the storage areas where the programs can manipulate. Each variable in GO programming has a specific type which is used to determine the size and layout of the variable's memory. So the default value of a local variable in GO programming language is "0".

Q22. What are the several built-in supports in Go?

GO has various built-in supports features in it, which help to perform it appropriately:
- Database
- Compression
- Web Server
- Container list and heap

Q23. What is Go programming?

Go is an open-source statically typed, compiled programming language for building simple, reliable, and efficient software. It was designed by **Robert Griesemer**, **Rob Pike**, and **Ken Thompson** at Google.

Please Visit OnlineInterviewquestions.com to download more pdfs