

# GO INTERVIEW QUESTIONS

[http://www.tutorialspoint.com/go/go\\_interview\\_questions.htm](http://www.tutorialspoint.com/go/go_interview_questions.htm)

Copyright © tutorialspoint.com

Dear readers, these **Go Programming Language Interview Questions** have been designed specially to get you acquainted with the nature of questions you may encounter during your interview for the subject of **Go Programming Language**. As per my experience good interviewers hardly plan to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer:

What is Go?

Go is a general-purpose language designed with systems programming in mind. It was initially developed at Google in year 2007 by Robert Griesemer, Rob Pike, and Ken Thompson. It is strongly and statically typed, provides inbuilt support for garbage collection and supports concurrent programming. Programs are constructed using packages, for efficient management of dependencies. Go programming implementations use a traditional compile and link model to generate executable binaries.

What are the benefits of using Go programming?

Following are the benefits of using Go programming –

- Support for environment adopting patterns similar to dynamic languages. For example type inference `x:=0` is *invalid declaration of a variable of type int*.
- Compilation time is fast.
- InBuilt concurrency support: light-weight processes *via goroutines*, channels, select statement.
- Conciseness, Simplicity, and Safety.
- Support for Interfaces and Type embedding.
- Production of statically linked native binaries without external dependencies.

Does Go support type inheritance?

No support for type inheritance.

Does Go support operator overloading?

No support for operator overloading.

Does Go support method overloading?

No support for method overloading.

Does Go support pointer arithmetics?

No support for pointer arithmetic.

Does Go support generic programming?

No support for generic programming.

Is Go a case sensitive language?

Yes! Go is a case sensitive programming language.

What is static type declaration of a variable in Go?

Static type variable declaration provides assurance to the compiler that there is one variable existing with the given type and name so that compiler proceed for further compilation without needing complete detail about the variable. A variable declaration has its meaning at the time of compilation only, compiler needs actual variable declaration at the time of linking of the program.

What is dynamic type declaration of a variable in Go?

A dynamic type variable declaration requires compiler to interpret the type of variable based on value passed to it. Compiler don't need a variable to have type statically as a necessary requirement.

Can you declared multiple types of variables in single declaration in Go?

Yes Variables of different types can be declared in one go using type inference.

```
var a, b, c = 3, 4, "foo"
```

How to print type of a variable in Go?

Following code prints the type of a variable –

```
var a, b, c = 3, 4, "foo"
fmt.Printf("a is of type %T\n", a)
```

What is a pointer?

It's a pointer variable which can hold the address of a variable.

For example –

```
var x = 5
var p *int
p = &x
fmt.Printf("x = %d", *p)
```

Here x can be accessed by \*p.

What is the purpose of break statement?

break terminates the for loop or switch statement and transfers execution to the statement immediately following the for loop or switch.

What is the purpose of continue statement?

continue causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.

What is the purpose of goto statement?

goto transfers control to the labeled statement.

Explain the syntax for 'for' loop.

The syntax of a for loop in Go programming language is –

```
for [condition | ( init; condition; increment ) | Range]
{
    statement(s);
}
```

Here is the flow of control in a for loop –

1. if condition is available, then for loop executes as long as condition is true.
2. if for clause that is *init; condition; increment* is present then

The init step is executed first, and only once. This step allows you to declare and initialize any loop control variables. You are not required to put a statement here, as long as a semicolon appears.

Next, the condition is evaluated. If it is true, the body of the loop is executed. If it is false, the

body of the loop does not execute and flow of control jumps to the next statement just after the for loop.

After the body of the for loop executes, the flow of control jumps back up to the increment statement. This statement allows you to update any loop control variables. This statement can be left blank, as long as a semicolon appears after the condition.

The condition is now evaluated again. If it is true, the loop executes and the process repeats itself *body of loop, then increment step, and then again condition*. After the condition becomes false, the for loop terminates.

3. if range is available, then for loop executes for each item in the range.

Explain the syntax to create a function in Go.

The general form of a function definition in Go programming language is as follows –

```
func function_name( [parameter list] ) [return_types]
{
    body of the function
}
```

A function definition in Go programming language consists of a *function header* and a *function body*. Here are all the parts of a function –

- **func** func starts the declaration of a function.
- **Function Name** – This is the actual name of the function. The function name and the parameter list together constitute the function signature.
- **Parameters** – A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.
- **Return Type** – A function may return a list of values. The **return\_types** is the list of data types of the values the function returns. Some functions perform the desired operations without returning a value. In this case, the return\_type is the not required.
- **Function Body** – The function body contains a collection of statements that define what the function does.

Can you return multiple values from a function?

A Go function can return multiple values. For example –

```
package main
import "fmt"
func swap(x, y string) (string, string) {
    return y, x
}
func main() {
    a, b := swap("Mahesh", "Kumar")
    fmt.Println(a, b)
}
```

In how many ways you can pass parameters to a method?

While calling a function, there are two ways that arguments can be passed to a function –

- **Call by value** – This method copies the actual value of an argument into the formal parameter of the function. In this case, changes made to the parameter inside the function have no effect on the argument.
- **Call by reference** – This method copies the address of an argument into the formal parameter. Inside the function, the address is used to access the actual argument used in the call. This means that changes made to the parameter affect the argument.

What is the default way of passing parameters to a function?

By default, Go uses call by value to pass arguments. In general, this means that code within a function cannot alter the arguments used to call the function and above mentioned example while calling max function used the same method.

What do you mean by function as value in Go?

Go programming language provides flexibility to create functions on the fly and use them as values. We can set a variable with a function definition and use it as parameter to a function.

What are the function closures?

Functions closure are anonymous functions and can be used in dynamic programming.

What are methods in Go?

Go programming language supports special types of functions called methods. In method declaration syntax, a "receiver" is present to represent the container of the function. This receiver can be used to call function using "." operator.

What is default value of a local variable in Go?

A local variable has default value as it corresponding 0 value.

What is default value of a global variable in Go?

A global variable has default value as it corresponding 0 value.

What is default value of a pointer variable in Go?

Pointer is initialized to nil.

Explain the purpose of the function Printf.

Prints the formatted output.

What is lvalue and rvalue?

The expression appearing on right side of the assignment operator is called as rvalue. Rvalue is assigned to lvalue, which appears on left side of the assignment operator. The lvalue should designate to a variable not a constant.

What is the difference between actual and formal parameters?

The parameters sent to the function at calling end are called as actual parameters while at the receiving of the function definition called as formal parameters.

What is the difference between variable declaration and variable definition?

Declaration associates type to the variable whereas definition gives the value to the variable.

Explain modular programming.

Dividing the program in to sub programs *modules/function* to achieve the given task is modular approach. More generic functions definition gives the ability to re-use the functions, such as built-in library functions.

What is a token?

A Go program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol.

Which key word is used to perform unconditional branching?

goto

What is an array?

Array is collection of similar data items under a common name.

What is a nil Pointers in Go?

Go compiler assign a Nil value to a pointer variable in case you do not have exact address to be assigned. This is done at the time of variable declaration. A pointer that is assigned nil is called a nil pointer. The nil pointer is a constant with a value of zero defined in several standard libraries.

What is a pointer on pointer?

It's a pointer variable which can hold the address of another pointer variable. It de-refers twice to point to the data held by the designated pointer variable.

```
var a int
var ptr *int
var pptr **int
a = 3000
ptr = &a
pptr = &ptr
fmt.Printf("Value available at **pptr = %d\n", **pptr)
```

Therefore 'a' can be accessed by \*\*pptr.

What is structure in Go?

Structure is another user defined data type available in Go programming, which allows you to combine data items of different kinds.

How to define a structure in Go?

To define a structure, you must use type and struct statements. The struct statement defines a new data type, with more than one member for your program. type statement binds a name with the type which is struct in our case.

The format of the struct statement is this –

```
type struct_variable_type struct {
    member definition;
    member definition;
    ...
    member definition;
}
```

What is slice in Go?

Go Slice is an abstraction over Go Array. As Go Array allows you to define type of variables that can hold several data items of the same kind but it do not provide any inbuilt method to increase size of it dynamically or get a sub-array of its own. Slices covers this limitation. It provides many utility functions required on Array and is widely used in Go programming.

How to define a slice in Go?

To define a slice, you can declare it as an array without specifying size or use make function to create the one.

```
var numbers []int /* a slice of unspecified size */
/* numbers == []int{0,0,0,0,0} */
numbers = make([]int,5,5) /* a slice of length 5 and capacity 5 */
```

How to get the count of elements present in a slice?

len function returns the elements presents in the slice.

What is the difference between len and cap functions of slice in Go?

len function returns the elements presents in the slice where cap function returns the capacity of

slice as how many elements it can be accomodate.

How to get a sub-slice of a slice?

Slice allows lower-bound and upper bound to be specified to get the subslice of it using[lower-bound:upper-bound].

What is range in Go?

The range keyword is used in for loop to iterate over items of an array, slice, channel or map. With array and slices, it returns the index of the item as integer. With maps, it returns the key of the next key-value pair.

What are maps in Go?

Go provides another important data type map which maps unique keys to values. A key is an object that you use to retrieve a value at a later date. Given a key and a value, you can store the value in a Map object. After value is stored, you can retrieve it by using its key.

How to create a map in Go?

You must use make function to create a map.

```
/* declare a variable, by default map will be nil*/  
var map_variable map[key_data_type]value_data_type  
/* define the map as nil map can not be assigned any value*/  
map_variable = make(map[key_data_type]value_data_type)
```

How to delete an entry from a map in Go?

delete function is used to delete an entry from the map. It requires map and corresponding key which is to be deleted.

What is type casting in Go?

Type casting is a way to convert a variable from one data type to another data type. For example, if you want to store a long value into a simple integer then you can type cast long to int. You can convert values from one type to another using the cast operator as following:

```
type_name(expression)
```

What are interfaces in Go?

Go programming provides another data type called interfaces which represents a set of method signatures. struct data type implements these interfaces to have method definitions for the method signature of the interfaces.

## What is Next ?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)

Loading [MathJax]/jax/output/HTML-CSS/jax.js