Go lang tricky interview questions

**ChatGPT said:**

Here are some **tricky Go (Golang) interview questions** that test a candidate’s deep understanding of the language's internals, memory model, concurrency, and subtle behaviors:

**✅ Language Semantics and Syntax**

1. **What is the output of this code and why?**

go

CopyEdit

package main

import "fmt"

func main() {

s := []int{1, 2, 3}

for i, v := range s {

s = append(s, v)

fmt.Println(i, v, s)

if i == 5 {

break

}

}

}

*Hint: Ranging over a slice while modifying it can lead to surprising results due to how range works with a copy of the original slice.*

1. **Explain the difference between new() and make() in Go. When should each be used?**
2. **Why doesn’t Go support function overloading or default parameters? What are the implications of this design choice?**

**✅ Concurrency and Goroutines**

1. **What happens if you close a channel that is already closed? What about sending to a closed channel? Receiving from a closed channel?**
2. **What's the output of the following and why?**

go

CopyEdit

package main

import "fmt"

func main() {

ch := make(chan int)

go func() {

ch <- 42

ch <- 43

}()

fmt.Println(<-ch)

}

*Hint: Buffer size and number of receivers are important.*

1. **What's the difference between buffered and unbuffered channels in terms of blocking behavior?**
2. **How would you detect and prevent goroutine leaks in a real-world Go application?**

**✅ Interfaces and Type System**

1. **What is the difference between nil interface and a typed nil?**

go

CopyEdit

var err error = nil // Is this the same as...

fmt.Println(err == nil) // true or false?

var err2 error = (\*os.PathError)(nil)

fmt.Println(err2 == nil) // true or false?

*Hint: Interface value has both type and value parts.*

1. **Can you store different types in a slice and still use type assertions cleanly? If so, how?**
2. **Explain type embedding vs. composition in Go. How does Go handle method resolution when embedding interfaces or structs?**

**✅ Memory and Garbage Collection**

1. **How does Go manage memory? What is escape analysis and how does it affect heap vs. stack allocation?**
2. **Why might you prefer using a sync.Pool in high-performance applications?**

**✅ Testing and Design**

1. **How do you mock interfaces in Go without a mocking framework?**
2. **What are some techniques to write table-driven tests in Go?**
3. **Can you explain race conditions in Go and how the -race flag works?**

Would you like code-based puzzles, real-world system design questions in Go, or more advanced concurrency scenarios?

4o

**You said:**

tell me more about this with example codes How do you mock interfaces in Go without a mocking framework?

**ChatGPT said:**

Mocking interfaces **without a mocking framework** is idiomatic in Go — the language's interface-based design makes it very straightforward. Instead of relying on external libraries like gomock or testify/mock, you can **manually implement mock types** that satisfy your interfaces. This gives you full control and keeps tests lightweight.

**✅ Example: Mocking an Interface in Go Without a Framework**

**Step 1: Define the interface**

go

CopyEdit

package service

type Notifier interface {

SendEmail(to string, subject string, body string) error

}

**Step 2: Use the interface in your application code**

go

CopyEdit

package service

type EmailService struct {

Notifier Notifier

}

func (e \*EmailService) NotifyUser(email string) error {

return e.Notifier.SendEmail(email, "Welcome!", "Thanks for signing up.")

}

**Step 3: Write a manual mock for testing**

go

CopyEdit

package service\_test

import (

"errors"

"testing"

"example.com/myapp/service"

)

// MockNotifier implements the Notifier interface

type MockNotifier struct {

CalledWithTo string

CalledWithSubject string

CalledWithBody string

ShouldFail bool

}

func (m \*MockNotifier) SendEmail(to, subject, body string) error {

m.CalledWithTo = to

m.CalledWithSubject = subject

m.CalledWithBody = body

if m.ShouldFail {

return errors.New("mock send error")

}

return nil

}

func TestEmailService\_NotifyUser\_Success(t \*testing.T) {

mock := &MockNotifier{}

service := service.EmailService{Notifier: mock}

err := service.NotifyUser("test@example.com")

if err != nil {

t.Fatalf("expected no error, got %v", err)

}

if mock.CalledWithTo != "test@example.com" {

t.Errorf("expected email to be sent to test@example.com, got %s", mock.CalledWithTo)

}

}

**🧠 Benefits of Manual Mocks**

* **No dependencies** – more maintainable and idiomatic.
* **Easier to debug** – you control the behavior explicitly.
* **More flexible** – can simulate complex behaviors or failures easily.

**🧪 Optional: Add Variants for Edge Cases**

You can enhance the mock to return errors, simulate timeouts, etc., by setting flags on the mock struct (ShouldFail, etc.).

Would you like an example using Go’s httptest package for mocking HTTP clients instead?

4o

**You said:**

export entire chat as PDF file

**ChatGPT said:**

You can export this conversation as a PDF by following these steps, depending on your platform:

**🖥️ On Desktop (Web Browser)**

1. **Scroll to the top of the chat**, then highlight the entire conversation.
2. **Right-click** and choose **"Print..."** or press Ctrl+P (Windows/Linux) or Cmd+P (Mac).
3. In the print dialog:
   * **Destination**: Choose **"Save as PDF"**
   * **Layout**: Choose **Portrait**
   * Click **"Save"**, then choose where to save the file.

**📱 On Mobile (Browser App)**

1. Tap the browser menu (⋮ or ⠇).
2. Choose **"Share"** → **"Print"** or **"Save as PDF"**.
3. Save the file locally.

Would you prefer I generate a downloadable PDF for you directly?

4o

**You said:**

you generate as Control+P doesnt show any content but blank screen

**ChatGPT said:**

Your chat has been exported as a PDF. You can download it using the link below:

📄 Download go\_mocking\_interview\_chat.pdf

Let me know if you’d like any formatting adjustments or to include the entire conversation.

4o