Slip 1

Q1 A) Write a program in GO language to accept user choice and print answers using arithmetic operators.

```
package main
import "fmt"
func main() {
 var a, b float64
 var choice int

 fmt.Print("Enter first number: ")
 fmt.Scanln(&a)
 fmt.Print("Enter second number: ")
 fmt.Scanln(&b)

 fmt.Println("Choose operation:")
 fmt.Println("1. Add")
 fmt.Println("2. Subtract")
 fmt.Println("3. Multiply")
 fmt.Println("4. Divide")
 fmt.Print("Enter choice: ")
 fmt.Scanln(&choice)

 if choice == 1 {
  fmt.Println("Result:", a+b)
 } else if choice == 2 {
  fmt.Println("Result:", a-b)
 } else if choice == 3 {
  fmt.Println("Result:", a*b)
 } else if choice == 4 {
  if b != 0 {
   fmt.Println("Result:", a/b)
  } else {
   fmt.Println("Cannot divide by zero!")
  }
 } else {
  fmt.Println("Invalid choice!")
 }
}
```

Slip 2

Q1 B) Write a program in GO language to print file information.

```
package main
import "fmt"
func main() {
 var n int
 fmt.Print("Enter the number of terms: ")
 fmt.Scanln(&n)
 a, b := 0, 1
 fmt.Println("Fibonacci Series:")
for i := 0; i < n; i++ {
}
}
fmt.Print(a, " ")
a, b = b, a+b
fmt.Println()
```

Slip 3

Q1 A) Write a program in the GO language using function to check whether accepts number is palindrome or not.

```
package main
import "fmt"
func isPalindrome(n int) bool {
rev, temp := 0, n
for temp > 0 {
rev = rev*10 + temp%10
temp /= 10
}
return n == rev
}
func main() {
var num int
fmt.Print("Enter number: ")
fmt.Scanln(&num)
fmt.Println(num, "is palindrome:", isPalindrome(num))
}
```

Slip 4

Q1 B) Write a program in GO language to sort array elements in ascending order.

```
package main
import (
"fmt"
"sort"
)
func main() {
arr := []int{5, 2, 8, 1, 3}
sort.Ints(arr)
fmt.Println("Sorted array:", arr)
}
```

Slip 5

Q1. A) Write a program in GO language program to create Text file

```
package main
import (
"fmt"
"os"
)
func main() {
f
 ile, err := os.Create("sample.txt")
if err != nil {
fmt.Println("Error:", err)
return
}
defer file.Close()
fmt.Println("File created successfully")
}
```

Slip 6

Q1 B) Write a program in GO language to copy all elements of one array into another using a method.

```
package main
import "fmt"
func copyArray(src []int) []int {
 dest := make([]int, len(src))
 copy(dest, src)
 return dest
}
func main() {
 arr1 := []int{1, 2, 3, 4, 5}
 arr2 := copyArray(arr1)
 fmt.Println("Copied array:", arr2)
}
```

Slip 7

Q1. B) Write a program in GO language to create structure student. Writea method show() whose receiver is a pointer of struct student.

```
package main
import "fmt"
func main() {
 var r, c int
 fmt.Print("Enter rows and columns: ")
 fmt.Scan(&r, &c)
 matrix := make([][]int, r)
 fmt.Println("Enter matrix:")
 for i := range matrix {
  matrix[i] = make([]int, c)
  for j := range matrix[i] {
   fmt.Scan(&matrix[i][j])
  }  }
 fmt.Println("Transpose:")
 for i := 0; i < c; i++ {
  for j := 0; j < r; j++ {
   fmt.Print(matrix[j][i], " ")
  }
  fmt.Println()
 }
}
```

Slip 8

Q1. A) Write a program in GO language to accept the book details such as BookID, Title, Author, Price. Read and display the details of ‘n’ number of books

```
package main
import "fmt"
func main() {
   var n int
   fmt.Print("Enter number of books: ")
   fmt.Scan(&n)
   books := make([]struct {
      ID     int
      Title  string
      Author string
      Price  float64
   }, n)
   for i := range books {
      fmt.Print("Enter BookID, Title, Author, Price: ")
      fmt.Scan(&books[i].ID, &books[i].Title, &books[i].Author, &books[i].Price)
   }
   fmt.Println("\nBook Details:")
   for _, b := range books {
      fmt.Println(b.ID, "\n", b.Title, "\n", b.Author, "\n", b.Price)
   }
}
```

Slip 9

Q1. A) Write a program in GO language using a function to check whether the accepted number is palindrome or not.

```
package main
import "fmt"
func isPalindrome(n int) bool {
 rev, temp := 0, n
 for temp > 0 {
  rev = rev*10 + temp%10
  temp /= 10
 }
 return n == rev
}
func main() {
 var num int
 fmt.Print("Enter number: ")
 fmt.Scanln(&num)
 fmt.Println(num, "is palindrome:", isPalindrome(num))
}
```

Slip 10

Q1. B) Write a program in GO language to read and write Fibonacci series to the using channel.

```
package main
import "fmt"
func fibonacci(n int, ch chan int) {
    a, b := 0, 1
    for i := 0; i < n; i++ {
        ch <- a
        a, b = b, a+b
    }
    close(ch)
}
func main() {
    ch := make(chan int)
    go fibonacci(10, ch)
     for num := range ch {
        fmt.Print(num, " ")
    }
}
```

Slip 11

Q1. A) Write a program in GO language to check whether the accepted number is two digit or not.

```
package main
import "fmt"
func main() {
 var n int
 fmt.Scan(&n)
 if n >= 10 && n <= 99 || n <= -10 && n >= -99 {
  fmt.Println("Two-digit number")
 } else {
  fmt.Println("Not a two-digit number")
 }
}
```

Slip 12

Q1. A) Write a program in GO language to swap two numbers using call by reference concept.

```
package main
import "fmt"
func swap(a, b *int) {
    *a, *b = *b, *a
}
func main() {
    var x, y int
    fmt.Println("Enter two numbers:")
    fmt.Scan(&x, &y)
    swap(&x, &y)
    fmt.Println("After swap:", x, y)
}
```

Slip 13

Q1. A) Write a program in GO language to print sum of all even and odd numbers separately between 1 to 100.

```
package main
import "fmt"
func main() {
 evenSum, oddSum := 0, 0
 for i := 1; i <= 100; i++ {
  if i%2 == 0 {
   evenSum += i
  } else {
   oddSum += i
  }
 }
 fmt.Println("Even Sum:", evenSum)
 fmt.Println("Odd Sum:", oddSum)
}
```

Slip 14

Q1. A) Write a program in GO language to demonstrate working of slices (like append, remove, copy etc.)

```
package main
import "fmt"
func main() {
s := []int{1, 2, 3}     // Create a slice
s = append(s, 4, 5)     // Append elements
fmt.Println("Slice:", s)
s = append(s[:1], s[2:]...) // Remove element at index 1
fmt.Println("After Remove:", s)
c := make([]int, len(s)) // Copy slice
copy(c, s)
fmt.Println("Copied Slice:", c)
}
```

Slip 15

Q1. A) Write a program in GO language to demonstrate function return multiple values.

```
package main
import "fmt"
func addSub(a, b int) (int, int) {
return a + b, a - b
}
func main() {
sum, diff := addSub(10, 5)
fmt.Println("Sum:", sum, ",Difference:", diff)
}
```

Slip 16

Q1. B) Write a program in GO language that prints out the numbers from 0 to 10, waiting between 0 and 250 ms after each one using the delay function.

```
package main
import (
"fmt"
"math/rand"
"time"
)
func main() {
rand.Seed(time.Now().UnixNano())
for i := 0; i <= 10; i++ {
fmt.Println(i)
time.Sleep(time.Duration(rand.Intn(250)) * time.Millisecond)
}
}
```

Slip 17

Q1. A) Write a program in GO language to illustrate the concept of returning multiple values from a function. (Add, Subtract, Multiply, Divide)

```
package main
import "fmt"
func calculate(a, b int) (int, int, int, float64) {
return a + b, a - b, a * b, float64(a) / float64(b)
}
func main() {
sum, diff, mul, div := calculate(10, 5)
fmt.Println("Sum:", sum, "Difference:", diff, "Multiplication:", mul, "Division:", div)
}
```

Slip 18

Q1. A) Write a program in GO language to print a multiplication table of number using function.

```
package main
import "fmt"
func table(n int) {
 for i := 1; i <= 10; i++ {
  fmt.Println(n, "x", i, "=", n*i)
 }
}
func main() {
 var num int
 fmt.Scan(&num)
 table(num)
}
```

Slip 19

Q1. A) Write a program in GO language to illustrate the function returning multiple values(add, subtract).

```
package main
import "fmt"
func addSub(a, b int) (int, int) {
   return a + b, a - b
}
func main() {
   sum, diff := addSub(10, 5)
   fmt.Println("Sum:", sum, ",Difference:", diff)
}
```

Slip 20

Q1. B) Write a program in Go language how to create a channel and illustrate how to close a channel using for range loop and close function.

```
package main
import "fmt"
func main() {
 ch := make(chan int) // Create a channel
 go func() {
  for i := 1; i <= 5; i++ {
   ch <- i // Send values to the channel
  }
  close(ch) // Close the channel
 }()
 for val := range ch { // Read values until channel is closed
  fmt.Println(val)
 }
}
```