Defer:-

A special statement in Go language which schedules a function to be called, whenever a particular function completes its execution.

It actually sequences a particular function after another function's execution is completed.

It is mainly used when resources need to be freed in one way or the other.

Eg:-

package main

import "fmt"

func first() { fmt.Println("1st")}

func second() { fmt.Println("2nd")}

func main() {

defer second()

first()

}

Explanation:- The above program will execute second() function only after the first one's execution is done.

Panic & Recover:-

Panic indicates an error in the program logic(For example:- panic would be called if we are trying to access an element of an array that is out of bounds,), or a condition from which it is not easy to recover from.

In order to handle Panic , there is a built-in RECOVER function which is used to stop the execution of panic function and returns value that was initially passed when panic occurred.

In order to recover from run time Panic, we pair the recover function with defer statement.(If we don't use defer statement then after Panic would take place the execution of program will stop and recover could not be invoked. )

package main

import "fmt"

func main() { defer func() {

str:= recover

fmt.Println(str) } ()

panic ("PANIC")

}

Here , we pair recover with defer so that as soon as panic executes, the program doesn't cease but due to defer function immediately recovery starts and the initial value is returned.