* What are defer, panic and recover? How do you recover from a run-time panic? (Page-91)

Defer is used to ensure that a specific function call is performed later after the program’s execution. Hence if you call that specific function using defer, it will be executed after all the exections in a program. Defer is commonly used to simplify functions that perform various clean-up actions. Defer functions are performed in a last-in-first-out manner.

In the below example,

func abc() {

for i=0; i<2; i++ {

defer fmt.Println(i)

}

}

It will print the output as 210.

Panic function can be caused by a run-time error. When a function calls panic, it stops executing and returns to the caller. The defer functions of the function work normally even in a panic.

Recover is a function that recovers a panic. When a function is panicking, the recover function will capture the value given to panic and will resume normal execution.

It is usual practice to use recover with defer while writing a panic statement, so as to be sure for the recover to work after a function has panicked.

function main() {

defer func () {

str := recover ()

fmt.Println(str)

}()

panic(“panic”)

}