Question 5:

Defer:

It pushes the function call in the list and makes sure that the function call is executed later in program execution. The deferred function is performed when the defer statement is executed. Defer is used when there are various functions to be performed to simply it.

e.g. If we create a file and write to it before closing it, we can defer the closing of that file. So closure will be done at the end of the program execution after all the writing is done.

Function main() {  
 file :=create(“test.txt”)

Defer close(f)

Write(f)

}

Panic: Panic points towards the programmer’s errors or exceptional condition. They can also be caused because of run-time errors. When the function calls panic, the executions of function gets stopped, program is terminated and error is reported with the panic value. Panic is used to abort such function which returns error, we don’t know to handle.

Recover: It is a built-in function which regains control from panic. It is useful only when used inside deferred function. Else, it will return null with no any other effect. Recover catched the error value of panic and resumes the execution of the function.

Recover from Run-time Panic:

package main

import "fmt"

func main() {

example()

fmt.Println("Finish")

}

func example(){

defer func() {

r := recover()

fmt.Println(r) }()

panic("Failure Panic")

}