Golang

two prospective dev and comp save money  
  
600 request memory less consuming 4kb for one routine and one thread lightweight  
  
  
define project   
workspace path where we are working  
path variable   
root variable compiler sit   
  
dependency mangment go.mod file  
  
packages smaller unit of project oragnizing our code   
  
constant and variable

const int a=1;

int a =1;

var name string  
  
datatypes int ,float,string, premitive datatypes  
  
slice and array  
slice linear structure   
array limited length

interface in golang  
non oop to oop interface is variable mocking

map key value pair data structure

non imports and imports

struct define with a small letter start is non import and if first letter is Capital than it can be importable to other files in package

goroutine  
two function runs concurrently same time in parallel depends on cores

5 functions running on 4 cores switching is very

parallelism vs concurrency  
1 is hardware dependent  
2 which achieve parrallelism by switching process very fast

go is built to use parrellism very efficinetly which is also called concuurrency

channles   
mechanism to communicate between goroutine

types of channels

buffer and unbuffered discard values

bidirectional unidirectional

differ mechanism work like try and catch of java execute at the end of program  
  
multi differ how it is execute stack based lifo

errors in golang

fatal error db connectivity error  
panic error api call   
  
anonyomous no name it is used write away it is defined

main func init func

main starting of program entry point

define prerequisite in init connect db url parametes

run project go run main.go  
go build to gen executable file

combining all dependency in one place is workspace

block chain development key points

solidity