* Panic Recover– Advance
  + How to get stack trace(runtime.Stack)
  + Cases that recovery is impossible
  + Detail program behavior after panic begin
* What is “...interface{}”
* Recursion – Intermediate
  + Difference beetwen c++(any) stack and golang stack
* Deferred Function Calls – Intermediate
  + sequence of calls defer(LIFO)
  + Closing file description
  + On-entry and on-exit actions
  + Accessing result variables
* Familiar with embedded interfaces behavior
* Netwroking
  + Create a transport to http.Get/Post
  + Able to write net server/clinet
  + TCP/UDP package processing experinese DNS for example
  + `gopacket` package expertise.
  + TCP/UDP client/server application development with bytes and gopacket operation.
  + Developing of TCP/UDP servers with specific requirements(latency, highload)
* Packages
  + Building Packages
  + go get and go get -u
  + go list
  + Workspace Organization
  + Internal Packages
* Struct
  + Multi composition
  + Method "inheritance" with composition
  + Fields alignment
  + custom tags
  + reflecting fields
  + structs conversion(was in types)
* Basic syntax
  + Operators
  + Understand how to handle errors and exceptions in Golang
  + Pointer tricks
  + Iterable types
  + Generics
* Concurrent Programming
  + GOMAXPROCS, GOSHED, NumCPU
  + Difference from the os thread
  + max number goroutines in OS
  + Goroutine Scheduling
  + Goroutine stack size
  + Channel – Advanced
    - Pipelines
    - Cancellation and broadcast mechanism

GO Lang General information

* How to use interactive shell
* The difference between commands go run, go build, go install
* The difference between environment variables $GOROOT, $GOPATH
* Golang workspace
* The difference between commands golint, go vet, gofmt
* What the command is goimports
* What the command is “go get”
* What variables is $GOOS, $GOARCH
* Advanced
  + Build constraints
  + Binary-Only Packages
  + How to install Golang from source
  + Versions of Golang (Existing versions)