Desired Learning Outcome

The learner will have demonstrated his/her strong understanding of the following topics.

## Golang Programming

We expect the learner to have understood and have accumulated some practical experience in the following topics related to the Golang programming language.

1. Understanding Go type system
   1. Constants, variables, numbers, empty variable
   2. Classes in OOP vs structures in Golang
   3. Arrays, slices and maps in Golang
2. Environment and Golang modules (go mod, GOPATH, …)
3. Understanding of the Go standard library
   1. String Operations (strconv, strings packages)
   2. Encoding and Binary (encoding/json, binary packages)
   3. Data layer (database/sql)
   4. Networking layer (net/http, io, net/rpc)
4. Dependency Injection in Go
   1. Understanding interfaces in Go vs OOP
   2. Understanding SOLID principles
5. Concurrency in Go
   1. Concurrency vs Parallelism
   2. Understanding Goroutines, OS Threads
   3. Understanding synchronisation primitives (sync, sync/atomic packages)
   4. Understanding channels in Go
6. Testing in Go
   1. Understand and able to write unit tests and end-to-end tests
   2. Benchmarking in Go

## Backend Programming

1. Able to create and design backend service APIs
   1. HTTP/1.1 REST (JSON encoding)
   2. Practical gPRC experience (protobuf encoding, HTTP/2)
   3. Authentication mechanisms (cookies, tokens)
   4. Basics of writing secure code (SQL Injection, XSS)
2. Database design (RDBMS)
   1. Understanding conceptual and relational models
   2. Understanding ACID
   3. Understanding RDBMS Transactions
   4. Understanding caching and database scaling

## Overall

The learner should be able to put everything together in a project and combine:

* Design a database schema
* Design an API contract for a service
* Able to implement a working service

Besides, as a backend software engineer at Grab, you have to possess the qualifications below:

* Excellent analytical and problem-solving skills, able to think critically
* Able to design, build, analyze and fix large-scale systems, also to debug and modify complex, production software
* Quickly learn full stack performance tuning and optimization