



## GO Programming

### Overview:

The GO programming language is among the most prolific open source & rare languages designed to take advantage of current age multi-core, distributed/cloud computing & network centric world. It is co-created by the same distinguished people who created C programming language & Unix which are among the greatest contributions to computer science, keeping in mind next generation requirements.

GO programs combine the flexibility of dynamically typed language with the safety & efficiency of statically typed and compiled language. Working with GO is fast & it produces executables which can run on all key platforms including Windows, Linux/Unix (many flavours), Mac OS X.. GO has “batteries included” - providing a whole set of useful libraries & tools one may need. Learning GO is definitely among the best investments one can make.

This hands-on program from Cralina is designed to give a through introduction to working with GO.

### Target Audience

- Software Engineers, Developers/Designers, Architects, Managers
- Anyone who wants to learn designing/programming in GO language

### Pre-requisite

- Familiarly with programming and operating systems

### Takeway

- Solid understanding of GO language concepts
- Good primer on how to write enterprise grade code using GO language

### Delivery Method:

Instructor lead and hands on with suitable examples/production code. The program has solid emphasis on hands-on coding. Lots of code we write in this program may be used as boilerplate code for enterprise projects.

### Program Contents:

#### *Introduction to GO*

- Why learn GO
- More than “Hello World”
- Standard Library



### *Program Structure*

- Naming, declaration, variables, scoping etc.

### *GO Packages and tools*

#### *Data types*

- Basic Data Types
- Composite Data Types

#### *Functions*

- Declaration, Anonymous, Variadic, Deferred calls, Panic & Recovery

#### *Error Handling*

- Understanding Error Handling, recommended ways to handle errors in GO

#### *Methods*

- Working with methods,
- Composing types

#### *Working with Data*

- *Handling JSON*

#### *Interfaces*

- Polymorphism and more

#### *Concurrency*

- Parallelism, Race Conditions, Mutex, Channels Preview

#### *Routines and Channels*

- Introduction, Understanding Channels and Goroutines

#### *Dependency Management*

- Overview, handling OS level dependencies

#### *Benchmarking, Testing*

- Using GO tools to benchmark & test programs

*Best practices will be covered with every topic.*

**Duration:** 2.5 days (20 hours)