

**Advanced Go Programming**

|  |  |  |  |
| --- | --- | --- | --- |
| **Duration** | 2 Days | **Modality** | Virtual / ILT |
| **Platform** | Windows, Linux, MacOS | **Level** | Intermediate |

**Overview**

Go Language (golang) is an open source programming language and has become increasing popular as a general-purpose language. You can develop a wide variety of applications using Go, such as systems, database, and networking applications. The combination of the unique capabilities of the language and enumerable features has made Go the preference for developing innovative exciting new applications.

Learn professional programming from a Go Programming expert. Create a modern application using the tools provided in this class. This course provides students the practical tools to build a complete Go application. At the end of this course, students will be able to leverage the more advanced features of Go to create fully-functional and sophisticated applications. Gain important hands-on and real-world experience in the labs at the end of each module. You will create applications in class that you can leverage elsewhere.

This is a hands-on course with several exercises and labs. You will write Go source code from the beginning to end of class. You will garner lots of experience in class.

**Audience**

The audience are individuals with three months prior development or scripting experience with Go. Attending the Go Foundation class also fulfills the prerequisite.

This class is available for a variety of platforms: Windows, Linux, or MacOs. You can also use your preferred integrated development environment.

**Solutions**

Emerge from class creating secure and networked applications. Learn to create Go applications with advanced features, such as atomicity, memory management, and unsafe capabilities.

**What you learn:**

1. Concurrency
   1. Concurrency concepts
   2. Goroutines
   3. WaitGroup
   4. Channel
2. Files
   1. JSON
   2. Marshall/Unmarshal
   3. Streaming
   4. Customization
3. Cryptography
   1. Public / Private Key
   2. Encryption
   3. Hashing
   4. Digital Signatures
4. Unit Testing
   1. Creating a unit test
   2. Go test command
   3. Initialization and cleanup
   4. Sub tests
   5. TestMain
5. Reflection
   1. Types
   2. General inspection
   3. Reflecting values
   4. Dynamic calls
6. Modules
   1. Packages
   2. Directory hierarchy
   3. Modules
   4. Versioning
7. Concurrency Synchronization
   1. Channel
   2. Locks
   3. Reader Writer Locks
   4. Cond
   5. Mutex
8. C2Go
   1. Go to C
   2. C to Go
   3. Marshaling
   4. Dynamic memory allocation
   5. Embedding C code