

# Kurs-Dokumentation



**Zentrum für Informatik ZFI AG**

**Google Go für C++ Programmierer (golang for  
c++ programmers) (GOCP)**

<http://www.zfi.ch/GOCP>

Weitere Infos finden Sie unter [www.zfi.ch](http://www.zfi.ch) oder via Adresse:

**Zentrum für Informatik ZFI AG  
Zentralsekretariat  
Rütistrasse 28  
CH-8952 Zürich-Schlieren  
Telefon: 044 732 40 00  
Telefax: 041 530 31 68**

**Zürich, Basel, Bern, Zürich, Schweiz**

<b>Titel</b>	<b>Google Go für C++ Programmierer (golang for c++ programmers)</b>
<b>Untertitel</b>	<b>Eine effiziente Einführung in die Programmierung mit Google Go für C++ Programmierer</b>
<b>Einleitung</b>	<p>Die Programmiersprache Google Go hat sich seit der offiziellen Freigabe im Jahr 2009 zur Programmiersprache des Jahres entwickelt. Immer mehr wird Google Go auch in grösseren Projekten eingesetzt.</p> <p>Go ist eine pragmatische Programmiersprache und verbindet Konzepte von C,Java und Python. Go Programme sind zudem sehr schnell, da solche native kompiliert werden und damit wird kein zusätzliches Environment benötigt. Go ist zudem portabel und läuft auf allen gängigen Plattformen.</p> <p>Von der Sprache her ist Go streng typisiert, bietet aber keine Klassen sondern Strukturen mit Methoden und unterscheidet sich damit von C++. So ist z.B. auch die Vererbung mit Go nicht möglich und dies ganz bewusst, denn es bietet die Komposition von Typen und auch Interfaces. Go Programme sind damit zwingend schlanker aufgebaut. Go arbeitet analog C mit Pointern, aber ohne deren Arithmetik.</p> <p>Mit Google Go lassen sich parallele Abläufe mit Goroutines schnell und einfach programmieren. Go bietet reiche Concurrency Features und dies ist eines der wertvollsten Elemente dieser Sprache.</p> <p>Go wird zur Zeit vor allem serverseitig eingesetzt und enthält einen integrierten Web Server (Http). Damit lassen sich auch Web Applikationen effizient und einfach programmieren.</p> <p>Durch den einfachen Aufbau der Sprache bietet Go zu allen Aufgaben eine ähnliche Lösung und ist damit weniger vielfältig, dies ist ein enormer Vorteil für die Arbeit in Teams. Man kann den Programmcode leichter verstehen.</p> <p>Mit diesem Kurs steigen Sie als C++ Developer schnell in den Google Go Zug ein. Wir arbeiten neben der Theorie vor allem mit Codebeispielen und Übungen. Wir referenzieren immer wieder auch die Quellen im Internet, so dass Sie umfassend informiert sind.</p> <p>Mit dem Jumpstart beginnen wir zudem gleich mit dem Schreiben erster Unit Tests. Testing und Benchmarks sind direkt in Google Go integriert und lightweight.</p> <p>Dieser Google Go Kurs vermittelt die Konzepte anhand von gezielt gewählten Beispielen und Anwendungsfällen. Damit folgt der Kursaufbau den pragmatischen Zielen von Go. Die Theorie wird direkt im Beispielcode oder als Notizen vermittelt. So verfügen Sie nach dem Kurs über umfassende Use Cases von Go.</p>
<b>Ihr Nutzen</b>	<ul style="list-style-type: none"><li>- Sie kennen die Ziele und Architektur von Google Go</li><li>- Sie kennen die zentralen Sprachkonzepte von Google Go im Vergleich mit C++</li></ul>

	<ul style="list-style-type: none"><li>- Sie können Google Go Programme schreiben, builden und ausführen, oder einfach gesagt: "Sie können mit Go programmieren"</li><li>- Sie wissen wie man grössere Google Go Programme strukturiert und erstellt.</li><li>- Sie kennen die wichtigsten Quellen im Internet</li></ul>
<b>Voraussetzungen</b>	<b>Erfahrung mit der C++ Programmiersprache</b>
<b>Teilnehmerkreis</b>	<b>C++ Programmierer und Anwender</b>
<b>Unterlagen</b>	<ul style="list-style-type: none"><li>- Tutorial</li><li>- Internet / Intranet</li><li>- Begleitbuch</li></ul>
<b>Folgekurse</b>	<b>Google Go Concurrent Programming, Kurscode GOCO</b>
<b>Inhalt</b>	<ul style="list-style-type: none"><li>- Einführung und JumpStart</li><li>- Installation und IDE</li><li>- Pakete und Imports</li><li>- Funktionen</li><li>- Variablen und Scope</li><li>- Basistypen, Typumwandlungen, Typinferenz, Nullwerte, Konstanten, Auto Increment (iota)</li><li>- Entscheidungen mit if und switch</li><li>- For-Schleife</li><li>- Pointer und Strukturen</li><li>- Arrays und Slices</li><li>- Maps</li><li>- make und new</li><li>- Funktionen und Closures</li><li>- Verzögerte Ausführungen mit defer</li><li>- Naming Conventions, Exportables</li><li>- Panic und Recover</li><li>- Methoden und Strukturen</li></ul>

- Interfaces
- Stringer Interface
- Objektorientierte Programmierung mit Go und Interfaces
- Fehlerbehandlung
- Reflection
- Das io-Packet (Reader und Writer)
- Goroutinen (schlanke Threads) und Kanäle
- Synchronisieren mit sync.Mutex
- Package Testing, Unit Testing
- Benchmarks
- Subtests, Sub-benchmarks
- Diverses

**Beitrag**

Der Teilnehmerbeitrag versteht sich rein netto. Das ZFI ist (gemäss MwSt-Gesetz) nicht Mehrwertsteuerpflichtig und erhebt somit keine MwSt. Bei länger als einen Monat dauernden Lehrgängen ist die Zahlung des Teilnehmerbeitrages in mehreren Raten möglich (pro rata temporis).