

# Kurs-Dokumentation



## Zentrum für Informatik ZFI AG

### Grundlagen der Programmiersprache C++ (CBAS)

<http://mobile.zfi.ch/CBAS>

Weitere Infos finden Sie unter [mobile.zfi.ch](http://mobile.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, ZÄ¼rich, Schweiz

<b>Titel</b>	<b>Grundlagen der Programmiersprache C++</b>
<b>Untertitel</b>	<b>eine Einführung</b>
<b>Einleitung</b>	<b>C++ ist eine moderne Programmiersprache, die alle aktuellen Paradigmen unterstützt. Sie ist heute aus der systemnahen Programmierung und auch aus weiten Teilen der Anwendungsprogrammierung nicht mehr wegzudenken. Zahlreiche praktische Übungs-Beispiele vermitteln dem Teilnehmenden eine gewisse erste Routine im Erstellen von C++-Programmen.</b>
<b>Ihr Nutzen</b>	<b>Die Teilnehmenden werden befähigt, C++-Programme selber zu erstellen und zu warten. Sie kennen die Ablauf- und Datenstrukturen von C++ und können selbständig Fehler lokalisieren. Sie kennen die Grundsätze der objektorientierten Programmierung.</b>
<b>Voraussetzungen</b>	<b>Kenntnisse einer anderen Programmiersprache wie Java, Visual Basic, Basic, COBOL, Delphi, Pascal, C, Assembler, PL/I etc. Für Programmier-Anfänger ist der vorgängige Besuch des Kurses "Programmier-Grundkurs" (NFGL) unerlässlich ! Kenntnisse der englischen Sprache sind von Vorteil.</b>
<b>Teilnehmerkreis</b>	<b>Personen, welche die Sprache C++ erlernen möchten, um damit eigene Programme erstellen zu können.</b>
<b>Unterlagen</b>	<b>ZFI-Kursordner</b>
<b>Folgekurse</b>	<b>- "C++ Advanced" (CADV)- "Embedded C++" (ECP)- "OO Grundlagen mit UML" (OUGL)- "OO Design" (ODES)</b>
<b>Inhalt</b>	<b>- Einführung in C++ - Ein- und Ausgabe - cout - cin  - Einfache Erweiterungen in C++ - Kommentar - Blockkonzept - Auflösungsoperator - Komma-Operator - Präfix und Postfix - Strukturen in C++ - Union in C++ - bool - string - dynamische Speicherverwaltung - new - delete - Fehlerbehandlung über new-Handler  - Erweiterungen bei Funktionen - inline-Funktionen - Prototyping - Defaultwerte</b>

- Variable Parameterlisten
- Referenzen und Referenzvariablen
- Referenzparameter
- Referenzen als Funktionsergebnisse
- Überladen von Funktionen
  
- Klassen
- Klassenkonzept
- Instanzen einer Klasse
- Zugriffsattribute
- Konstruktoren und Destruktoren
- Friends
- Statische Klassenelemente
- Geschachtelte Klassen
- Die Attribute mutable und explicit
  
- Vererbung
- Strukturen und Klassen
- Konstruktoren und Destruktoren abgeleiteter Klassen
- Zuweisungskompatibilität in Klassenhierarchien
- Klassenhierarchien
- this-Zeiger
- Virtuelle Methoden
- Virtuelle Destruktoren
  
- Mehrfachvererbung
- Grundlagen
- Virtuelle Basisklassen
- Zugriffsrechte bei Vererbung und Überladung
  
- Polymorphismus
- Rein virtuelle Methoden
- Abstrakte Basisklassen
  
- Datei- und String-Streams
- Datei-Streams
- String-Streams
  
- Überladen von Operatoren
- Overloading durch friend-Funktionen
- Overloading mit Methoden
- Überladen der Post- und Präfix-Operatoren
- Typumwandlungs-Operatoren
- Konstruktoren als Typumwandlungs-Operatoren
- Typumwandlungsoperator-Funktionen

- Kopieren von Objekten
- Überladen des Funktionsoperators ()
- Überladen des Operators ->
- Überladen von new und delete
  
- Namensbereiche
- Namensbereiche
- std
  
- Templates
- Funktions-Templates
- Klassen-Templates
  
- Standard-Template-Library
- STL-Begriffserklärung
- Iteratoren
- vector
- deque
- list
- set und multiset
- map und multimap
- string
- Algorithmen der STL
  
- Exception-Handling
- Verschachtelte Exception-Blöcke
- eigene Exception-Klassen definieren
  
- Objektorientierter Entwurf
- Einführung

**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).