

# Moderne C++-Programmierung («CPROGR»)

C++ hat sich in den letzten Jahren sehr verändert und ist deutlich mächtiger geworden. Neue Sprachfeatures erlauben eine einfachere und somit sicherere Entwicklung. Diese Vorteile kommen immer mehr auch im Embedded Umfeld zum Einsatz. Machen Sie sich fit!

**Dauer:** 4 Tage

**Preis:** 3'200.- zzgl. 8.1% MWST

**Kursdokumente:** Digicomp Kursmaterial und Begleitbuch

## Inhalt

Der Kurs besteht aus den folgenden Modulen. Die Gewichtung und Auswahl der Module wird den Kenntnissen und Präferenzen der Teilnehmer angepasst.

1. Basisfeatures
  - Namespace
  - Referenzen
  - Überladen von Funktionen
2. Wichtige Standard-Klassen
  - Verwendung von Streams
  - String
  - Vektor
3. Klassen
  - Klassendefinition und Instanzierung
  - Attribute (Datenelemente)
  - Zugriff auf Klassenelemente
  - Methoden (Elementfunktionen)
  - Information Hiding
  - Konstruktor
4. Module und Schnittstellen
  - Deklaration und Definition
  - Inline-Funktionen
  - Statische Klasselemente
5. Klassenableitung, Vererbung
  - Vererbung und Sichtbarkeit der Elemente
  - Initialisierung von Basisklassen
  - Vererbung und Polymorphismus
6. Dynamische Speicherverwaltung
  - new-Operator
  - delete-Operator
  - Dynamische Instanzen verwenden
7. Normalform für Klassen
  - Copy-Konstruktor
  - Deep Copy
  - Kopieren verhindern
  - Destruktor
  - Zuweisungsoperator
  - Move, Rvalue-Referenzen
8. Smart-Pointer
  - unique\_ptr
  - shared\_ptr
9. Virtuelle Methoden

- Statisches Binden
- Dynamisches Binden
- Virtuelle Methoden
- Abstrakte Klassen

#### 10. Ein- und Ausgabe in Dateien

#### 11. Templates

- Templates definieren
- Templates anwenden
- Funktionstemplates

#### 12. Die Standard Template Library (STL)

#### 13. Container

- vector, deque
- list
- set
- map

#### 14. Funktionen und Funktionsobjekte

- Lambda Expression
- Prädikat, Gleichheit und Äquivalenz

#### 15. Algorithmen

#### 16. Exceptions

- throw / try / catch
- Exception-Hierarchien

## Key Learnings

- Einsetzen von Klassenbibliotheken in Ihren Programmen
- Verstehen und Nutzen von Scopes und Namespaces
- Einsetzen der string-Klasse für die Handhabung von Zeichenketten
- Definieren von Klassen mit den zugehörigen Attributen und Methoden
- Verstehen von Klassen als ein Hilfsmittel für Information Hiding
- Implementieren und Einsetzen von Konstruktoren
- Sinnvolles Modularisieren von Code
- Definieren einer Klassenhierarchie
- Kennen der verschiedenen Sichtbarkeiten von Klasselementen in Bezug auf die Ableitung
- Anwender der Operatoren new und delete
- Erstellen eines Programms mit dynamischer Speicherverwaltung
- Kennen der Vorteile von Smartpointern
- Verstehen von *unique\_ptr* und *shared\_ptr*
- Definieren von virtuellen Methoden für dynamisches Binden
- Definieren von Schnittstellen mit rein virtuellen Funktionen und abstrakten Klassen
- Mit Streams auf Dateien zugreifen
- Grundkenntnisse zu Templates
- Korrekte Anwendung der Container *vector*, *deque*, *list*, *set*, *map*, *unordered\_set*
- Effizientes Suchen und Sortieren von Daten dank den STL-Algorithmen
- Verstehen der Funktionsweise der Fehlerbehandlung mit Exceptions

## Zielpublikum

C/C++-Programmierer, die objektorientierte C++-Programme entwickeln möchten. Projektleiter im C++-Umfeld, die C++ produktiv und effizient einsetzen wollen

## Anforderungen

Sie müssen mindestens die Grundlagen der Sprachen C oder C++ beherrschen. Wir empfehlen Ihnen Grundkenntnisse der objektorientierten Programmierung entsprechend den folgenden Kursen:

- [Grundlagen der Sprachen C und C++ \(«CPG»\)](#)

## Haben Sie Fragen oder möchten Sie einen Firmenkurs buchen?

Wir beraten Sie gerne unter 044 447 21 21 oder [info@digicomp.ch](mailto:info@digicomp.ch). Detaillierte Infos zu den Terminen finden Sie unter [www.digicomp.ch/weiterbildung-software-engineering/programmiersprachen/c-c/kurs-moderne-c-programmierung](http://www.digicomp.ch/weiterbildung-software-engineering/programmiersprachen/c-c/kurs-moderne-c-programmierung)