



Programmieren mit C und C++ - Lehrgang (CPLLG)

C++ stellt dem Programmierer eine Vielfalt an Datenstrukturen und Algorithmen zur Verfügung, die einfach da sind und funktionieren. Dem Programmierer werden Sprachkonstrukte — und ein geniales Typensystem — zur Verfügung gestellt, um eigene Libraries zu erstellen. Dass C++ eine einfache Programmiersprache ist, hat allerdings noch keiner gewagt zu behaupten.

Dieser 10-tägige Lehrgang besteht aus zwei Modulen, beginnend mit dem „Professionellen Einstieg in die Programmiersprache C“ und dem anschließenden Modul „C++ Grundlagen“, in welchem die Teilnehmer fundiertes Wissen über die Sprache C++, Sprachmittel und Libraries erlangen. Bei Buchung dieses Ausbildungspakets profitieren Sie darüber hinaus von einem erheblichen Preisvorteil zu den Einzeltrainings.

INHALTE

MODUL 1: Programmieren mit C - Grundlagen

Grundlagen: C ist eine Sprache, um nackten Speicher zu manipulieren. Maschineninstruktionen werden direkt ausgeführt. Das bringt einen erheblichen Geschwindigkeitsvorteil gegenüber interpretierten Sprachen — aber auch den Nachteil, dass man zu jeder Zeit genau wissen muss, was man tut. In diesem Kapitel werden Grundlagen erklärt, die für das Verständnis der Speicherverwaltung eines Programmes zu tun haben.

Kontrollkonstrukte: Das wenigste, worin sich C von anderen Sprachen unterscheidet, sind Kontrollstrukturen wie if, switch, while, sowie Funktionen.

Pointer sind das Beliebteste an C, da sie eine Unmenge an Flexibilität einbringen. Sie sind auch das Unbeliebteste an C, da sie eine Unmenge an Fehlermöglichkeiten einbringen. Neben Pointern werden auch Möglichkeiten gezeigt, die unvermeidlichen Fehler zu finden.

Implizite Typumwandlungen. Ein Hauptkritikpunkt an C ist die sorglose Konvertierung zwischen signed und unsigned Integertypen, zwischen Integertypen verschiedener Breite, und zwischen Floating Point Zahlen und Integers. Die Konvertierungen folgen nichtsdestotrotz einem Regelwerk, das man verstanden haben sollte, um korrekte Programme schreiben zu können.

Der C-Präprozessor: So leid es uns tut, er ist Teil der Sprache, und muss zumindest in Grundzügen gezeigt werden.

Modularisierung ist heutzutage aus der Programmierung nicht mehr wegzudenken. Ein Grund für den Erfolg von C war damals die Möglichkeit, Teile von Programmen separat zu bauen und sie erst

PREIS P. P.

€ 3990,- (zzgl. MwSt.)

DAUER

10 Tage (09:00 - 17:00 Uhr)

SIE HABEN FRAGEN?

Infocenter WIEN: +43 1 934 9 934
Infocenter GRAZ: +43 316 422 966

E-Mail Anfrage: office@tectrain.at

<https://www.tectrain.at/seminare/c-programmierung-basisausbildung>





später zu verbinden. Hier gibt es einiges zu erzählen, und leider spielt der Präprozessor eine nicht zu kleine Rolle.

Verschiedenes:

- Debugginghilfen wie das `assert()` Macro
- Sichtbarkeit, oder die verschiedenen Bedeutungen des Wortes `static`
- Die Toolchain: Compiler und Linker, Optimierungsmöglichkeiten, ...
- Analyse von Memoryfehlerszenarien

Die Standard-Library

- **Dynamischer Speicher:** `malloc()` und `free()`, und verwandte Funktionen, um explizit Speicher zu verwalten.
- **Standard I/:** Formatierte und komfortable Ein- und Ausgabe ist nicht gerade eine der Stärken von C. C hat mächtige Werkzeuge der Ein- und Ausgabe, jedoch sind diese weit weg von Komfort und Fehlerimmunität — schon alleine deswegen verdienen sie mehr als nur eine Erwähnung.
- **Stringbearbeitung:** Strings sind in C aus heutiger Sicht etwas eigenwillig definiert. Die Standard-Library bringt eine Fülle von Funktionen mit, die Stringbearbeitung erleichtern. Hier lauern wieder Gefahren (alle Arten von Memoryfehlern).

MODUL 2: Programmieren mit C++ - Grundlagen

Datenkapselung

- Klassen
- Konstruktoren und Destruktoren
- Kopieren von Objekten (inklusive Fallstricke)

Funktionen und Methoden

- Overloading
- `const`: der Weg zur Korrektheit
- Referenzen: Pointer mal anders
- `static`: die 100ste Bedeutung dieses Keywords
- Operator-Overloading

Exception Handling: Strukturierte Fehlerbehandlung, abseits von `if-else`

Templates: eine Einführung in generische Programmierung

Standard Template Library (STL)

- Sortieren
- Suchen



- Sequentielle Container
- Assoziative Container

Dynamische Speicherverwaltung: die Operatoren new und delete

Vererbung

- Grundlegende Mechanismen
- Interfaces — das virtual Keyword
- Mehrfachvererbung (inklusive Fallstricke)

Objektorientiertes Design: Grundlagen — alles andere wäre zuviel

Die Neuerungen aus C++11 und C++14: teilweise werden bereits in den bisherigen Kapiteln die Unterschiede zwischen “herkömmlichen” C++ (C++03) und dem “neuen” C++ erläutert. Es gibt mehr:

- Range Based for Loops
- Lambda Expressions
- rvalue Referenzen und Moving
- Threading

ZIELGRUPPE

Softwareentwickler

VORAUSSETZUNGEN

Praktische Erfahrung in einer beliebigen Programmiersprache.