

## *Einführung in C/C++*

*Form* Inhouse-Schulung

*Zeitungfang* 3 - 5 Tage

*Zielgruppe* Programmierer (C/C++-Anfänger oder Umsteiger)  
Software-Entwickler, Projektleiter

*Vorkenntnisse* Die freie und kreative Umgang mit formalen Modellierungen sollte beherrscht werden.  
Konkrete Programmiererfahrung ist nicht notwendig, aber von Vorteil.

*Ziele* Die Teilnehmer lernen die Programmiersprache C/C++ in ihrem systematischen Aufbau und den zugrunde liegenden Konzepten kennen.  
Der Kanon der Syntax, der Operatoren und Kontrollstrukturen wird im Detail angeeignet.  
Das Paradigma der 'objektorientierten Programmierung' (OOP) in seinen Motivationen und seine Umsetzung in C++ wird verstanden und als praktisch einsetzbare Kompetenz eingeübt.  
Zudem werden Systematik und Nutzen der C++-Standard-Bibliothek verfügbar gemacht.

*Gegenstand* Die Programmiersprache C++ verbindet modernste objektorientierte Programmierung mit der Geschwindigkeit und Freiheit von maschinennahem C. C++ ist die bewährte Basis für das erfolgreiche Design zukunftssicherer und performanter Software jeglicher Skalierung. C/C++ ist 'alt', weil die Sprache von Anfang an ein konsistenter Entwurf war. C/C++ ist hochmodern, weil es sich millionenfach in der Praxis bewährt und mit Bedacht weiterentwickelt wird. (Version 2011 ist aktuelle Schulungsgrundlage) C++ ist das Ideal- und Universalwerkzeug für den kompetenten Programmierer, der z.B. mit modernen Frameworks wie Qt5 komplexe Applikationen entwickelt und betreut.

*Kursinhalte*

- Geschichte und Konzept der Compiler-Sprache
- Typisierung, Syntax, Operatoren, Kontrollstrukturen, Funktionen
- Zeiger, Arrays und Zeichenketten
- Präprozessor-Direktiven
- Ein-/Ausgaben, Stream-Begriff, Dateizugriffe (je C und C++)
- Typ-Komposition mit Strukturen
- Motivation und Umsetzung der Objektorientierung
- Klassen, Namensräume und Schutzstrategien
- Static-Membervariablen und -Funktionen
- Überladung von Konstruktoren
- Operatoren-Überladungen
- Vererbung und Mehrfachvererbung
- Polymorphie, Abstraktion und RTTI
- Exception und strukturierte Fehler-Behandlung
- generische Programmierung (Funktionen, Klassen)
- C++-Bibliothensklassen (u.a. string, IO-Klassen)
- STL und generische container-Klassen
- diverse Erweiterungen/Neuheiten in C++11

(Der Stoff wird ggf. nach Zeit und Interesse angepaßt.)