MICRO CONSULT

Stand 11.01.2026

Saubere, effiziente und robuste C++-Software entwickeln mit klassischer Objektorientierung und modernen C++ Sprachmitteln - Der Weg zum Clean Code - Live-Online-Training

Ziele -
Ihr Nutzen

Sie beherrschen die wichtigsten "Best Practices", um sauberen, performanten und robusten C++-Code neu zu erstellen oder Legacy-Code in einem Refactoring-Projekt zu modernisieren.

Teilnehmer

Software-Entwickler/innen, Software-Architekt/innen

Voraussetzungen

Für die aktive Teilnahme am Praktikum im Workshop-Stil: Sie kennen bereits die Grundlagen der Objektorientierten Programmierung und verfügen über solide Grundkenntnisse in C++. Sie sind vertraut im Umgang mit einer gängigen Entwicklungsumgebung, die C++20 unterstützt. Für die Übungen kann ein eigener Laptop genutzt werden oder Sie arbeiten via Internet Bowser in einem bereitgestellten Online-Workspace.

Optional: Sollten Sie bisher kaum C++ Programmierkenntnisse haben, können Sie darauf verzichten, in den Übungen selbst zu programmieren und sich stattdessen auf einen "Vorher - Nachher" Vergleich des jeweiligen Schritts in Form eines reinen Code-Reviews konzentrieren.

Live Online Training

02.02. – 04.02.2026	1.950,00 €	3 Tage

* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: L-OOPFC++

Präsenz-Training - Deutsch

Termin	Dauer
20.04. – 22.04.2026	3 Tage

Präsenz-Training - Englisch

Dauer
5 Tage

Saubere, effiziente und robuste C++-Software entwickeln mit klassischer Objektorientierung und modernen C++ Sprachmitteln - Der Weg zum Clean Code - Live-Online-Training

Inhalt

Paradigmen, Muster, Idiome und Best Practices

- Guter Code von Anfang an
- Guter Code durch Refactoring
- Mögliche Motivationsprobleme
- Versteckte Kosten von schlechtem Code

- Zielkonflikte und Abwägungen

Grundregeln

- Wiederholungen vermeiden
- Offenheit für Erweiterungen
- Komplexität eingrenzen
- Modularisierung
- Testautomatisierung
- Optimierung nur mit klaren Zielen

C++-spezifische Aspekte

- Header- und Implementierungsdateien
- Sprache vs. Bibliotheken und Frameworks
- Vor- und Nachteile von Templates
- Welche Rolle spielt die Metaprogrammierung?

Nützliches in der Standard-Bibliothek

- Kurzüberblick von C++98 zu C++23 STL - Container, Iteratoren, Algorithmen
- "std::any", "std::variant"
- "std::tuple" und "structured binding"
- "std::function"

Klassische Objektorientierung mit C++

- Kapselung von Daten und Funktionen
- Geheimnisprinzip und Zugriffsschutz
- Vererbung, Interfaces, dynamisches Binden
- Komposition oft besser als Vererbung
- "SOLID" Prinzipien und mehr (SLA, YAGNI, …)

Kommunikation zwischen Software-Modulen

- Synchrone und asynchrone Verfahren
- Prinzip der losen Kopplung und hohen Kohäsion
- Performance-Steigerung durch Callbacks
- Vergleich und Bewertung typischer Alternativen (Linker, Zeiger/Referenzen, Interfaces, Templates)

Praktische Übungen im Workshop Stil

- Projektbasierte Kursstruktur Die praktischen Übungen sind als fortlaufendes Mini-Projekt konzipiert, das sich durch das gesamte Training zieht und aufeinander aufbaut. Jede Einheit dauert etwa 90 bis 120 Minuten, in denen Sie die Bearbeitungsgeschwindigkeit an Ihre individuellen Vorkenntnisse in C++ anpassen können.
- Unterstützung und Lösungen: Damit Sie stets auf dem richtigen Weg bleiben, stehen Ihnen alle Musterlösungen im Voraus zur Verfügung. So können Sie sicher sein, dass Sie den Anschluss nicht verlieren, auch wenn Sie einmal auf Herausforderungen stoßen. Der Trainer steht jederzeit für Fragen und Hilfestellungen zur Verfügung und unterstützt Sie während der Übungen.
- Zusätzliche Herausforderungen: Für fortgeschrittene Teilnehmende bietet jeder Schritt zusätzliche, optionale "offene Challenges", um den praktischen Teil des Kurses auch für erfahrene Programmierer/innen spannend und anspruchsvoll zu gestalten.

MicroConsult PLUS

- Den Code und die Beschreibung der Workshop-Übung erhalten Sie bereits vor dem Training, sodass Sie sich schon vorab mit den zu bearbeitenden Schritten sowie den offenen Challenges vertraut machen können.