Seminare & Workshops 2019

QA AKADEMIE
Wissen für bessere Software

Pole-Position für bessere Software

Modern C++1x Design

07./08.10.2019

Fastware

09./10.10.2019

Immer schneller als Ihre Bugs

Schwieberdinger Str. 56 | 70435 Stuttgart | seminare@qa-systems.de | www.qa-systems.de | Tel. +49 711 138183-0 | Fax +49 711 138183-10

Modern C++1x Design

Das sagen die Teilnehmer

"The topic is very complex, yet Andrei manages to explain it in a clear and entertaining way."
D. Germann, Plexim GmbH

"If only I could apply everything I learned here in my everyday work! Thank you!"
D. Osswald, Schunk GmbH & Co. KG

"Exceptionally Good! The whole style was perfect. I learned a lot of new things and the personal stories will help me to memorize the learned."
D. Feurle, sodgeIT GmbH

"Dr. Alexandrescu is excellent. He explains things in an easy way and understandable. I like his speech very much and from the class I learned lots of tricks and skills and deeper understanding to C++. Thank you!"
D. Jostmeyer, Bachmann Technology GmbH & Co.KG

"I liked at most, that example code was taken from real projects, there were good examples and Andrei explains really good. Thank you!"
C. Willbold, CipSoft GmbH

"Andrei's vast knowledge on C++ combines greatly with his humour making this course a must-see."
S. Montellese, Supercomputing Systems AG

Modern C++1x Design

Referent	Dr. Andrei Alexandrescu
Termin	7./8. Oktober 2019 Maritim Hotel, Ulm
Uhrzeit	9:00 Uhr – 17:00 Uhr
Dauer	2-tägig, englisch
Teilnahmegebühr	2.190,- € (zzgl. MwSt.)
1. Frühbucherpreis (20%)	1.752,- € (zzgl. 19% MwSt.) bei Anmeldung bis 31.03.2019
2. Frühbucherpreis (10%)	1.971,- € (zzgl. 19% MwSt.) bei Anmeldung bis 05.07.2019

Änderungen vorbehalten / changes reserved

The book "Modern C++ Design" popularized what became colloquially known as modern C++, a refreshing style of high-level programming that gave away inheritance and polymorphism in favor of templates and compile-time metaprogramming. Over time, support for this way of writing code has been significantly enhanced in C++11 and C++14. This seminar discusses the core topics of Modern C++ Design in the context of C++1x. With better language support certain artifacts are much easier to design and implement. More importantly, new opportunities are now available.

Course Highlights

Participants will gain:

- A good understanding of Design Patterns and their place in an engineer's toolchest.
- Mastery of policy-based design that uses C++ generic programming to implement highly reusable designs.
- A comprehensive introduction to C++1x core language features and library artifacts specifically targeted at template-based programming.
- A good understanding of classic design patterns, their tradeoffs, and state-of-the-art generic C++ implementation for each.

Outline

Please note: This course is being actively developed. The actual course might contain more topics and slight variations on the topics outlined below.

- Recap of Design Patterns
- Recap of Policy-Based Design
- Variadic Templates
- std::tuple
- Deferred std::tuple expansion
- Typelists and policy parameters
- Factory Design Pattern
- Visitor Design Pattern
- Chain of Responsibility Design Pattern
- Declarative Control Flow
- Unifying Exception with Error Codes
- Policy-Based Memory Allocation
- Generic Locking
- Robert Martin's Principles of Object-Oriented Design
  - SRP: The Single Responsibility Principle
  - OCP: The Open Closed Principle
  - LSP: The Liskov Substitution Principle
  - ISP: The Interface Segregation Principle
  - DIP: The Dependency Inversion Principle
  - REP: The Release Reuse Equivalency Principle
  - CCP: The Common Closure Principle
  - CRP: The Common Reuse Principle
  - ADP: The Acyclic Dependencies Principle
  - SDP: The Stable Dependencies Principle
  - SAP: The Stable Abstractions Principle

Attendee Profile

This class targets senior engineers and architects of C++-based systems. Familiarity with the major features of C++ is required. Template programming and C++1x concepts will be introduced.

Format

The format is a highly interactive lecture. Questions during the lecture are encouraged. Use of laptops for trying out examples is allowed.

Das sagen die Teilnehmer

"The explanations of Andrei led to many "Aha!" moments and made it definitely worth my time."

S. Montellese, Supercomputing Systems AG

"Thanks to Andrei for sharing his knowledge!"

P. Kaczmarczyk, ESG Elektroniksystem - und Logistik GmbH

"Bringing my application to speed of light seems possible after this course."

C. Lang, Supercomputing Systems AG

"Awesome!"

T. Böhme, IBM Deutschland Research & Development GmbH

"Andrei is the fast track to performance C++ know How."

D. Hummel, treibauf AG

Fastware

Referent	Dr. Andrei Alexandrescu
Termin	9./10. Oktober 2019 Maritim Hotel, Ulm
Uhrzeit	9:00 Uhr – 17:00 Uhr
Dauer	2-tägig, englisch
Teilnahmegebühr	2.190,- € (zzgl. MwSt.)
1. Frühbucherpreis (20%)	1.752,- € (zzgl. 19% MwSt.) bei Anmeldung bis 31.03.2019
2. Frühbucherpreis (10%)	1.971,- € (zzgl. 19% MwSt.) bei Anmeldung bis 05.07.2019

Änderungen vorbehalten / changes reserved

This two-day course introduces attendees to a thorough approach to optimization techniques for contemporary computing architectures. It is based on Andrei's career-long experience with tuning performance of various software systems, from Machine Learning research to high-performance libraries to Facebook-scale computing backends.

A large category of applications have no boundaries on desired speed, meaning there's no point of diminishing returns in making code faster. Better speed means less power consumed for the same work, more workload for the same data center expense, better features for the end user, better machine learning, better analytics, and more. Yet information on writing fast code is scant and difficult to find. Software engineering folklore is rife with tales of optimizations. Programmers commonly discuss and argue whether a piece of code is supposed to be faster than another, or what to do to improve the performance of a system small or large. The course teaches systematic, scientific approaches to measuring and improving code performance.

Optimization has received increased attention during the past decade, a trend that is likely to intensify. Serial execution speed has stalled and, after parallelizing what's possible, in any system we have single-thread speed of sheer algorithmic execution as the lasting bottleneck. Systematic algorithmic improvements that work in conjunction with hardware tradeoffs are key to improving performance. Optimization has always been an art, and in particular optimizing code on contemporary hardware has become a task of formidable complexity. This is because modern hardware has a few peculiarities about it that are not sufficiently understood and explored. This class offers a thorough dive in this fascinating world.

Outline

Please note: This course is being actively developed. The actual course might contain more topics and slight variations on the topics outlined below.

- The Art of Benchmarking
  - Amdahl's and Lhadma's Law
  - The use of intuition
- Conducting Time Measurements
  - Noise and how to avoid it
  - Quantization noise
  - Background noise
  - Overhead noise
  - Performance counters
- Baselines
  - Pitfalls in choosing baselines (or failing to)
  - Choosing good baselines
  - Differential timing
- Strength Reduction
  - Motivation
  - Strength operations hierarchy
  - Running example
- Minimizing Indirections
  - Motivation
  - Running example

Fastware

› Replacing branches with arithmetic
- Motivation
- Running example
- "Too much of a good thing"

› Eager Computation: Tables vs. Computation
- Dynamically initialized static data vs. statically initialized static data
- Running example

› Lazy Computation
- Memoization
- Computation vs. Tables
- Lazy Structuring

› "The Forgotten Parallelism:" Instruction-Level Parallelism
- Relationship with data dependencies
- Relationship with associativity of operations and other arithmetic underpinnigs
- Redesigning algorithms for minimizing data dependencies

› Inlining
- Beware compiler's most vexing inlining
- "Dark matter" code
- I-Cache considerations
- Controling inlining of cdtors
- Case study: a custom shared_ptr
  - Beware inline destructors
  - Lazy refcount allocation
  - Skip last decrement
  - Prefer zero
  - Use dedicated allocators
  - Use smaller counters

› Copy Elision
- Elision vs. moving
- API design to avoid unnecessary work

› Scalable Use of the STL
- A few words on hash tables
- The affix scrambling trick
- Resizing std::vector

› Building Structure on Top of Arrays
- Motivation
- The rule of "sevenish"
- Sorted arrays, and improvements thereof
- Binary heaps
- Bring to front
- Randomized bring to front

› Large Set Operations and Derivatives
- Bilinear version
- Improved version for skewed statistics
- Galloping search

› Contention Minimization
- Work on the side, lock, swap
- Readers-Writer locks
- Using R-W locks with the STL

› Algorithm design topics
- Sentinels
- Vacancies
- Faster Hoare partition
- Redesigning Quickselect
  - Better Layout
  - Smaller groups
  - Adaptation

Attendee Profile

This is aimed at C++ programmers who have efficiency of generated code as a primary concern.

Format

The format is a highly interactive lecture. Questions during the lecture are encouraged. Use of laptops for trying out examples is allowed.

Modern C++1x Design | Fastware

❍	**Modern C++1x Design**	
	7./8. Oktober 2019	2.190,- € (zzgl. MwSt.)
	1. Frühbucherpreis (20%)	1.752,- € (zzgl. 19% MwSt.) bei Anmeldung bis 31.03.2019
	2. Frühbucherpreis (10%)	1.971,- € (zzgl. 19% MwSt.) bei Anmeldung bis 05.07.2019
❍	**Fastware**	
	9./10. Oktober 2019	2.190,- € (zzgl. MwSt.)
	1. Frühbucherpreis (20%)	1.752,- € (zzgl. 19% MwSt.) bei Anmeldung bis 31.03.2019
	2. Frühbucherpreis (10%)	1.971,- € (zzgl. 19% MwSt.) bei Anmeldung bis 05.07.2019
	Paketpreis bei Buchung beider Seminare	3.285,- € (zzgl. MwSt.)
	1. Frühbucherpreis (20%)	2.628,- € (zzgl. 19% MwSt.) bei Anmeldung bis 31.03.2019
	2. Frühbucherpreis (10%)	2.956,50 € (zzgl. 19% MwSt.) bei Anmeldung bis 05.07.2019

Fax Anmeldung
+49 711 138183-10

Kontaktdaten des Teilnehmers

Firma

Name, Vorname

Abteilung

Straße

PLZ/Ort

Tel. / Fax

E-Mail (bitte angeben)

Rechnungsanschrift (falls abweichend)

Firma

Name, Vorname

Abteilung

Straße

PLZ/Ort

Bestellnummer (falls vorhanden)

Hotelreservierung: Wir haben Zimmerkontingente in verschiedenen Hotels zum Abruf für die Teilnehmer eingerichtet.
Bei Interesse senden wir Ihnen in der Bestätigungsemail gerne eine Übersicht mit allen Details dieser Abrufkontingente zu.
Sollte dies erwünscht sein, bitte hier ankreuzen:

❍ Hotelinformationen erwünscht

Datum/rechtsgültige Unterschrift

AGB für Seminare

Anmeldebestätigung
Nach Anmeldung erhalten Sie vorab eine Eingangsbestätigung Ihrer Anmeldung per E-Mail. Die offizielle Anmeldebestätigung mit Rechnung erhalten Sie rechtzeitig vor Seminarbeginn per Post.

Ort und Veranstaltungsdauer
Ort und Dauer der Veranstaltung erfahren Sie in den einzelnen Seminarbeschreibungen.

Preise/Zahlungskonditionen
Die Rechnungsstellung erfolgt vor Seminarbeginn. Die Rechnung ist nach Erhalt ohne jeden Abzug sofort, jedenfalls vor Seminarbeginn, fällig. Alle Preise verstehen sich zuzüglich der gesetzlichen Mehrwertsteuer. Im Übrigen gelten die gesetzlichen Bestimmungen des allgemeinen Zahlungsverkehrs. In den Seminargebühren sind die begleitenden Seminarunterlagen, Getränke sowie das Mittagessen enthalten.

Stornierung durch Teilnehmer
- bis 12 Wochen vor Veranstaltung kostenlos
- bei 12 – 4 Wochen vor Veranstaltung berechnen wir 20% der Teilnahmegebühr
- ab 4 Wochen vor Veranstaltung berechnen wir die volle Teilnahmegebühr

Der Austausch durch andere Personen ist jederzeit möglich!

Stornierung durch Veranstalter
Wir bitten um Verständnis, dass wir uns Absagen aus organisatorischen Gründen vorbehalten. Durch die Bestätigung der Anmeldung entsteht kein Rechtsanspruch auf die Durchführung des Seminars. QA Systems kann bei nicht erreichter Mindestteilnehmerzahl oder bei Hindernissen, die außerhalb des Einflusses liegen, Seminarveranstaltungen absagen. In Ausnahmefällen (z. B. Krankheit des Referenten oder höhere Gewalt) kann die Absage auch kurzfristig erfolgen. Wir bemühen uns in diesen Fällen unverzüglich um einen Ersatztermin. Bei einer Absage durch QA Systems werden bereits bezahlte Seminargebühren voll zurückerstattet.

Darüber hinausgehende Rechtsansprüche, insbesondere die Erstattung der Kosten aus Arbeitsausfall, Reise- oder Hotelkosten etc. bestehen nicht.

Hotelreservierung
QA Systems reserviert im jeweiligen Seminarhotel – in der Regel zu Sonderkonditionen – eine begrenzte Zimmeranzahl als Abrufkontingent. Die Reservierung sowie die Abrechnung der Zimmer sind von Ihnen direkt mit dem Hotel abzuwickeln.

QA SYSTEMS
The Software Quality Company

Schwieberdinger Str. 56 | 70435 Stuttgart | seminare@qa-systems.de | www.qa-systems.de | Tel. +49 711 138183-0 | Fax +49 711 138183-10