

# BIO

## Angelika Langer

**Trainer / Mentor / Consultant**



### Biography

I work as an independent freelance trainer, mentor, and consultant with a course curriculum of my own and conduct seminars in Europe and Northern America. My current work is backed by more than a decade of practical experience as a software engineer and team leader in Germany and the US.

Currently my preferred fields of interest are training, coaching, and mentoring in the area of object-oriented and component-based software development in C++ and Java. I invest a considerable amount of my precious time on development of explanatory material, including course material, multimedia training, books, and articles.

In addition, I conduct lectures, seminars, and workshops, mostly in Europe and North America. I enjoy speaking at OO conferences all over the world. I organize open enrollment courses in collaboration with partner companies in Germany, Switzerland, the Netherlands, the UK and the USA and conduct on-site training world-wide on demand.

The training business leaves me little time for consulting; there is simply no time left for extended periods of contractor work. The consulting jobs that I accept are mainly mentoring assignments including code reviews, audits, design sessions, special purpose workshops, etc.

### Course Curriculum

Currently I teach C++ and Java related topics based on my own course material. All courses are relatively advanced; I rarely teach introductory courses although I can make introductions to C++ and Java available on request. My curriculum includes Effective C++, Effective STL, Effective Java, Concurrent Programming in Java, Template Programming in C++, Exception Handling in C++, Stream I/O in C++. For further information see <http://www.langer.camelot.de/Courses.htm>.

### Publications

The list of my publications is too long to be mentioned in its entirety here. For a complete list see <http://www.langer.camelot.de/Publications.htm>.

In brevity: I am co-author of the widely recognized book "Standard C++ IOSTreams and Locales" published in 2000 by Addison-Wesley. I was a columnist of the US magazines C++ Report and C/C++ Users Journal for more than 4 years. I still function as a reviewer for Fawcette Technical Publications online magazine DevX. I have been publishing articles in various other magazines including the German magazines ObjectSpektrum and JavaSpektrum. Currently I consider writing another book on advanced Java topics.

### Public Lectures

I have been speaking at more conferences world-wide than I can remember. For a complete list see <http://www.langer.camelot.de/Conferences.htm>.

Past lectures were held at OOPLSA (USA), TOOLS (USA, Australia, Switzerland), OOP (Germany), ROOTS (Norway), C++ World (USA), Software Development (USA), DevWeek (UK), ACCU (UK), Application Development (UK), and several others.

### Contact Info

Angelika Langer  
Training & Consulting  
Object-Oriented and Component-Based Software  
Development in C++ & Java

Neumarkter Straße 86d  
81673 München, Germany  
phone: +49-89-436 38 54  
email: [langner@camelot.de](mailto:langner@camelot.de)  
URL: <http://www.langer.camelot.de/>

# Introduction To Java

- *Sound Introduction to Language, Tools and APIs*
- *Avoiding Traps and Pitfalls*
- *Covers J2SE 1.5 (incl. Generics)*

## Course Description

This course focuses on giving you an understanding of exactly what Java is and how to build, compile, and distribute effective stand-alone Java applications. The course is designed to be a comprehensive overview of the Java language and runtime model for developers experienced with another programming language. Upon completion, you will be able to program using Java. You will know how to create Java applications.

## Audience

Professional programmers who want to learn Java.

## Prerequisite

To fully benefit from this course, attendants should be experienced computer programmers who understand the rudiments of both procedural and object-oriented programming.

## Duration

5 days

## Language

English or German

## Course Overview

### Basic Java Language Features

- Java SDK
- Language Basics
- Arrays

### Classes

- Classes
- References
- Access Modifiers
- Static Fields and Methods

### Inheritance

- Inheritance
- Dynamic Binding

### Initialization

- Construction
- Final Variables

### Interfaces and Enumerations

- Interfaces
- Enumeration Types

### Error Handling

- Exception Handling
- Assertions

### Object Infrastructure

- Class Object
- Clone
- Equals

### Parameterized and Nested Types

- Parameterized Types
- Nested Types

### Reflection

- Reflection

### Utilities

- Collections
- I/O

### Concurrent Programming

- Threads
- Synchronization
- Concurrency Utilities

### Java Platform Library APIs

- Swing
- Networking
- JDBC

### Tools

- JavaDoc
- Jar

## Location

Open enrollment courses are conducted in collaboration with local partner companies. Public courses are offered at regular intervals.

The course schedule can be found at <http://www.langer.camelot.de/Courses/Schedule.html>.

Courses can also be held at your company site. Duration and content will then be tailored to your specific needs and prerequisites.

## Contact

For availability and enrollment send e-mail to [langer@camelot.de](mailto:langer@camelot.de). For further information check out <http://www.langer.camelot.de/Courses.html>.

# New Features in J2SE 5.0

- *Java 5.0 Language Features*
- *Java Generics*
- *Concurrency Utilities*

## Course Description

The 5.0 release of the Java 2 Standard Edition (J2SE) adds a number of major and minor new language features and library extensions. The most significant addition to the Java programming language are parameterized types and methods, a feature known as Java Generics. The most substantial addition to the Java platform libraries is the `java.util.concurrent` package that provides a set of powerful concurrency utilities.

## Objective

This course aims to **provides a comprehensive overview of the major and minor new features of the Java 5.0 release.**

## Audience

Professional programmers who want to keep their Java knowledge up to date.

## Prerequisite

The seminar should ideally be attended after some practical experience with Java.

## Duration

1 day

## Language

English or German

## Course Overview

### *Enums, Autoboxing, etc.*

- Enums
- Autoboxing
- Enhanced for-Loop
- Varargs

- Static Import
- Program Annotations

### **Generics**

- Generic Types and Methods
- Bounded Type Variables
- Translation by Type Erasure
- Use of Type Variables
- Use of Parameterized Types
- Automatic Type Inferences
- Wildcard Instantiations

### **Concurrency Utilities**

- Explicit Locks and Conditions
- Atomic Variables
- Runnable, Callable, Future
- Queue, Exchanger, Barrier, Latch
- Thread Pools
- Task Scheduling

## Location

Open enrollment courses are conducted in collaboration with local partner companies. Public courses are offered at regular intervals.

The course schedule can be found at <http://www.langer.camelot.de/Courses/Schedule.html>.

Courses can also be held at your company site. Duration and content will then be tailored to your specific needs and prerequisites.

## Contact

For availability and enrollment send e-mail to [langner@camelot.de](mailto:langner@camelot.de). For further information check out <http://www.langer.camelot.de/Courses.html>.

# Effective Java

- *Best Practice Programming Idioms*
- *Avoiding Traps and Pitfalls*
- *Advanced Language Features*

## Course Description

Java opens quite a number of trapdoors despite of its alleged reputation as an "easy-to-learn" programming language. For the more ambitious and professional Java programmer it is essential to understand the subtleties of the language and to know what to expect of Java, which language feature to use for which reason, and what to avoid in order to improve the quality of the resulting program.

## Objective

This course aims to shed some light on the more "interesting" areas of Java: it addresses pitfalls and helps avoiding common Java errors and it explains less commonly known, yet indispensable language features.

## Audience

Professional programmers who want to explore Java in greater depth.

## Prerequisite

The seminar should ideally be attended after some initial experience with Java and builds on elementary Java knowledge.

## Duration

4 days

## Language

English or German

## Course Overview

1. CONSTRUCTION AND FINALIZATION
  - Construction & Initialization
  - Destruction vs. Finalization

2. USE OF FINAL
  - Immutability (final + constness)
  - Function Redefinition (final + polymorphism)
3. OBJECT INFRASTRUCTURE
  - Object Equality and Comparison
  - Copying Objects (clone())
4. ABSTRACT CLASSES AND INTERFACES
  - The abstract Language Feature
  - Idioms for Use of Classes + Interfaces
5. INNER CLASSES
  - Language Features
  - Idioms for Use of Inner Classes
6. NEW LANGUAGE FEATURES
  - Enum Types, Annotations, Varargs
  - Autoboxing, Static Imports
7. GENERICS
  - Generic Types and Methods
  - Type Erasure and Wildcards
8. RTTI AND DYNAMIC PROGRAMMING
  - Type System and Reflection API
  - Dynamic Proxies
9. REFERENCES AND REACHABILITY
  - Reachability
  - Soft, Weak, and Phantom References
10. GARBAGE COLLECTION
  - Classic GC Algorithms
  - GC Tuning

## Location

Open enrollment courses are conducted in collaboration with local partner companies. Public courses are offered at regular intervals.

The course schedule can be found at <http://www.langer.camelot.de/Courses/Schedule.html>.

Courses can also be held at your company site. Duration and content will then be tailored to your specific needs and prerequisites.

## Contact

For availability and enrollment send e-mail to [lang@camelot.de](mailto:lang@camelot.de). For further information check out <http://www.langer.camelot.de/Courses.html>.

# High-Performance Java

- *Programming for Better Performance*
- *Avoiding Performance Pitfalls*
- *Profiling and Tuning Techniques*

## Course Description

Application performance is an important issue in every software development project. This seminar explores strategies for improving the performance of Java programs. The focus is on core Java – the language itself and its platform libraries (JDK). Attendants study programming techniques for efficient use of the Java language and its runtime environment, including tips for avoiding performance pitfalls. Benchmarking and profiling techniques are explained and practiced in hands-on labs. The goal is to enable attendants to identify and analyze performance bottlenecks and apply appropriate tuning techniques.

## Objective

This course aims to explain best practice programming techniques for high-performance-Java programs and practices performance profiling and analysis including appropriate tuning techniques.

## Audience

Professional programmers with an interest in producing high-performance Java software. The focus is on programming and tuning, not on high-level design and testing.

## Prerequisite

The seminar should ideally be attended by Java programmers with sound Java knowledge.

## Duration

2 days

## Language

English or German

## Course Overview

1. PERFORMANCE CONCEPTS
  - What is performance?
  - Performance and the Development Process
  - Performance Measuring
2. PROGRAMMING TECHNIQUES
  - a) **Programming**
    - Programming and Performance
    - Elementary Issues
    - Data Structures
    - I/O
    - Micro-Benchmarking
  - b) **Program Design**
    - Java Types and Performance
  - c) **Runtime Environment**
    - RAM Footprint
    - Class Loading
    - Garbage Collection
    - JVM Benchmarking
3. PROFILING, MONITORING & TUNING
  - a) **JVM Profiling**
    - Tool Architecture
    - Profiling Strategies and Tactics
    - Choosing a Profiler Tool
  - b) **GC Profiling**
    - Garbage Collections Algorithms
    - GC Profiling & Tuning Options

## Location

Open enrollment courses are conducted in collaboration with local partner companies. Public courses are offered at regular intervals.

The course schedule can be found at <http://www.langer.camelot.de/Courses/Schedule.html>.

Courses can also be held at your company site. Duration and content will then be tailored to your specific needs and prerequisites.

## Contact

For availability and enrollment send e-mail to [langner@camelot.de](mailto:langner@camelot.de). For further information check out <http://www.langer.camelot.de/Courses.html>.

# Concurrent Java

- *Multithread Programming in Java*
- *Synchronisation and Thread-Safety Issues*
- *Concurrency Utilities (since J2SE 1.5)*

## Course Description

Java supports concurrent programming with multiple threads directly by means of language features and library APIs. Java programmers therefore have to grasp the essence of concurrent programming no matter whether they actively create multiple threads or simply prepare their classes for safe use by multi threads. Professional Java programmers in general should be familiar with terms such as race conditions, synchronization, deadlocks, thread-safety, etc.

Since J2SE 1.5 the platform libraries offer a number of high-level abstraction that ease multithreaded programming in Java. This seminar explains the new abstractions.

## Objective

This seminar aims to give an introduction into concurrent programming using Java and exposes students to patterns and idioms that have been proven useful in practice.

## Audience

Software engineers who intend to build applications that are executed concurrently in multiple threads and programmers who need to prepare their classes for use in multi-threaded environments.

## Prerequisite

Sound knowledge of Java. Experience with concurrent programming helpful, but dispensable.

## Duration

3 days

## Language

English or German

## Course Overview

1. MULTI\_THREADING BASICS
  - Race Conditions
  - Thread Safety
  - Synchronization
  - Creating Threads
  - Runnable, Callable, Future
2. MULTI-THREADING IN ACTION
  - Lock-Free Programming
  - Nested Monitor Problem
  - Explicit Locks und Conditions
  - State-Dependent Operations
  - Thread Interruption
3. THE THREAD MODEL IN JAVA
  - Thread States and Scheduling
  - The Java Memory Model
  - Threads And Exceptions
  - Thread Local Memory
4. ADVANCED CONCURRENCY CONTROL
  - Queue, Exchanger, Barrier, Latch
  - Thread Completion (Join, Future, Callback)
  - Thread Pools
  - Task Scheduling

## Location

Open enrollment courses are conducted in collaboration with local partner companies. Public courses are offered at regular intervals.

The course schedule can be found at <http://www.langer.camelot.de/Courses/Schedule.html>.

Courses can also be held at your company site. Duration and content will then be tailored to your specific needs and prerequisites.

## Contact

For availability and enrollment send e-mail to [langner@camelot.de](mailto:langner@camelot.de). For further information check out <http://www.langer.camelot.de/Courses.html>.



# Reliable C++

- *Predictable Program Behavior*
- *Reliable Resource Management*
- *Object Validity and Const Correctness*

## Course Description

Software failures endanger lives, cost enterprises millions, and impair the software producer's credibility. If the reliability of your software is crucial to your business, if you aim for highest quality requirements, if you develop safety-critical components - then this is the right course for you.

This course does not cover the latest language features such as templates or exception handling. Instead, it aims for increasing your command the core language and provide you with skills to construct reliable, high-quality software. Topics include correct allocation and deallocation of memory, elimination of invalid pointers and memory leaks, efficient copying and assignment of objects, preservation of object validity, const correctness, redefinition of inherited member functions, and implicit type conversions in C++.

## Objective

In this course we explore strategies for achieving a higher degree of reliability and robustness of your C++ programs. It helps you to master the complexity of C++ and makes you aware of traps and pitfalls in the language.

## Audience

Software engineers for whom the reliability of their software is of paramount importance.

## Prerequisite

Knowledge of the language basics required, either gained in an introductory C++ course or equivalent practical experience.

## Duration

3 days

## Language

English or German

## Course Overview

1. **Managing the Lifetime of Objects**
  - Correct Initialization and Destruction
  - Safe Copying&Assignment of Objects
  - Destructors in Inheritance Hierarchies
  - Reference Counting
2. **Pointers and Memory Management**
  - Correct Allocation and Deallocation of Memory
  - Elimination of Invalid Pointers and Memory Leaks
3. **Data Access and Object Integrity**
  - Maintenance of Object Validity
  - Logical vs. Physical Constness to Internal Data
4. **Predictable Programs - How To Avoid Surprises**
  - Expression Evaluation
  - Sequence Points
  - Function Redefinition
  - Implicit Type Conversions

## Location

Open enrollment courses are conducted in collaboration with local partner companies. Public courses are offered at regular intervals in

- |                  |                 |
|------------------|-----------------|
| ▫ Amersfoort, NL | ▫ Stuttgart, DE |
| ▫ London, UK     | ▫ Boston, MA    |
| ▫ Zürich, CH     | ▫ Torrance, CA  |

Courses can also be held at your company site. Duration and content will then be tailored to your specific needs and prerequisites.

## Contact

For availability and enrollment send e-mail to [langner@camelot.de](mailto:langner@camelot.de). For further information check out <http://www.langer.camelot.de/Courses.htm>.

# Effective STL Programming

- *Using the Pre-Defined Standard Containers and Algorithms*
- *Extending the STL Framework by User-Defined Abstractions*
- *Getting Aware of Traps and Pitfalls in Using the STL*

## Course Description

The STL (Standard Template Library) is a comprehensive set of class and function templates for various collections and numerous algorithms and is part of the ISO/ANSI standard C++ library. It has an unusual and elegant architecture in that it separates data structures from algorithms - a programming paradigm known as generic programming.

This course you will learn how to effectively use the pre-defined collections and algorithms from the STL and how to extend the STL framework by adding special-purpose adaptors, iterators, containers, and algorithms. Along the way, you will be provided with tips and tricks for avoiding common errors. Moreover, the course does not only provide in-depth knowledge of the STL itself, but also explains new C++ language features and novel programming techniques that are used in the STL.

## Objective

The goal is not only to learn how to use the predefined components of the STL such as the standard containers and algorithms, but also understanding the underlying concepts and the design of the STL, including the principle of generic programming. "To use the STL is to extend it." Following this line of logic, we will also explore how to extend the STL framework by user-defined adaptors, iterators, containers, and algorithms.

## Audience

Software engineers who plan on using containers and algorithms from the standard C++ library.

## Prerequisite

Students should have a good command of the C++ programming language including basic knowledge of templates.

## Duration

4 days

## Language

English or German

## Course Overview

1. **Iterators**
  - Iterator concepts & iterator adapters
2. **Sequential containers**
  - Vector, deque & list
3. **Function objects**
  - Function adapters
4. **Associative containers**
  - Map and multimap & set and multiset
5. **Algorithms and Functors**
  - Functor requirements & concepts
6. **Containers In-Depth**
  - Container adapters & concepts
7. **Iterator In-Depth**
  - Iterator traits
  - Stream and insert iterators
8. **Smart Pointers**
  - auto\_ptr, scoped\_ptr, shared\_ptr
9. **Allocators**
  - Standard & user-defined allocators
10. **Exception safety**
  - The problem & safety guarantees

## Location

Open enrollment courses are conducted in collaboration with local partner companies. Public courses are offered at regular intervals in

- Amersfoort, NL
- Stuttgart, DE
- London, UK
- Zürich, CH

Courses can also be held at your company site. Duration and content will then be tailored to your specific needs and prerequisites.

## Contact

For availability and enrollment send e-mail to [langner@camelot.de](mailto:langner@camelot.de). For further information check out <http://www.langer.camelot.de/Courses.htm>.



# C++ Templates In Depth

- *Programming with C++ Templates*
- *Compile-Time Computations*
- *Generic Programming*

## Course Description

Traditionally, templates are used for implementation of parameterized types such as collections. This kind of usage is still useful, but C++ templates allow more than just that. Today, template programming has become a paradigm of its own. Compile-time computations, generic programming, and template meta-programming are only some of the buzz words in the area of template programming.

This course gives a comprehensive overview of all aspects of template programming, from templates-related language features over template-based programming techniques to a brief introduction into advanced and special-purpose techniques such as expression templates and template meta-programming. The focus, however, is on template programming techniques and idioms that are applicable to everyday problems. Such idioms include parameterized inheritance, generic conversions, and a discussion of runtime polymorphism vs. compile-time polymorphism.

## Objective

In this course we explore the template language feature in depth and learn how templates can be used effectively in your day-to-day practice for solving common programming problems in an efficient, elegant, and maintainable way.

## Audience

Software engineers for whom runtime performance of their programs is of paramount importance.

## Prerequisite

Knowledge of the language basics required, either gained in an introductory C++ course or equivalent practical experience.

## Duration

3 days

## Language

English or German

## Course Overview

1. **Basic Template Features**
  - Class and Function Templates
  - Template Parameters
  - Template Specialization
  - Member Templates
2. **Advanced Template Features**
  - Friends and Templates
  - Template Instantiation
  - Template Argument Deduction
  - Specializing and Overloading
3. **Template Programming Techniques**
  - Traits and Policy Classes
  - Static Virtuality
  - Template Metaprogramming
  - Expression Templates
4. **Advanced Applications**
  - Type Traits
  - Smart Pointers
  - Function Objects

## Location

Open enrollment courses are conducted in collaboration with local partner companies. Public courses are offered at regular intervals in

- |                  |                   |
|------------------|-------------------|
| ▫ Amersfoort, NL | ▫ Los Angeles, CA |
| ▫ London, UK     | ▫ Stuttgart, DE   |
| ▫ Zürich, CH     | ▫ Boston, MA      |

Courses can also be held at your company site. Duration and content will then be tailored to your specific needs and prerequisites.

## Contact

For availability and enrollment send e-mail to [langner@camelot.de](mailto:langner@camelot.de). For further information check out <http://www.langer.camelot.de/Courses.htm>.

# C++ Exception Handling In Depth

- *Catching and handling exceptions*
- *Exception-safe programming*
- *Object integrity and resource management*

## Course Description

Exception handling (EH) is the standard technique for indicating and handling errors in C++ programs. It provides a uniform way of reporting errors and makes up for several weaknesses of traditional error reporting techniques. EH adds a whole new dimension of programming techniques to C++ and has a significant impact not only on programming itself, but also on program design.

In this course, we dig deeper into exception handling issues and discuss questions such as: When are our programs exception-safe or exception-neutral? What level of safety guarantee do our functions give? Raising exceptions is easy, but coping with exceptions thrown at us is difficult. How can we make sure that our functions work reliably and predictably even in presence of exceptions? How does EH affect our design?

## Objective

This course aims to equip attendants with greater awareness of EH issues in general and how its affects class design and implementation in particular.

## Audience

Software engineers for whom runtime performance of their programs is of paramount importance.

## Prerequisite

Knowledge of the language basics required, either gained in an introductory C++ course or equivalent practical experience.

## Duration

1 day

## Language

English or German

## Course Overview

1. **Throwing Exceptions**
  - Stack Unwinding and Implicit Destruction
  - Grouping of Exceptions in Hierarchies
2. **Catching and Handling Exceptions**
  - Handler Sequences
  - Throw vs. re-throw
3. **Programming With Exceptions**
  - Avoid Resource Leaks
  - Resource Acquisition is Initialization
  - auto\_ptr , Smart Pointers, Scope Guards
  - Exceptions in Constructors
  - Exceptions in new Expressions
  - Exceptions in Destructors
  - Safety Guarantees
4. **Design of Exceptions**
  - Standard Exception Classes
  - Designing Exception Hierarchies
5. **Program Termination**
  - Uncaught Exceptions
  - Terminate- and Uncaught-Handler
6. **Exception Specifications**
  - Unexpected Exceptions
  - The Unexpected-Handler

## Location

Open enrollment courses are conducted in collaboration with local partner companies. Public courses are offered at regular intervals in

- |                  |                 |
|------------------|-----------------|
| ▫ Amersfoort, NL | ▫ Stuttgart, DE |
| ▫ London, UK     | ▫ Boston, MA    |
| ▫ Zürich, CH     | ▫ Torrance, CA  |

Courses can also be held at your company site. Duration and content will then be tailored to your specific needs and prerequisites.

## Contact

For availability and enrollment send e-mail to [langner@camelot.de](mailto:langner@camelot.de) . For further information check out <http://www.langer.camelot.de/Courses.htm>.

# C++ IOStreams In Depth

- *Using and Extending the Standard C++ IOStreams Library*
- *Internationalization in Standard C++: Locales and Facets*
- *Based on the book "Standard C++ IOStreams and Locales" by Angelika Langer & Klaus Kreft*

## Course Description

Formatting and parsing of text is a problem that many C++ programmers still solve by resorting to C functions like `printf()` and `scanf()`, despite of their indisputable drawbacks. A more reliable and less error prone tool for handling of text I/O is available in form of the C++ IOStreams library. Closely related to IOStreams are locales and facets, which form the standard C++ support for internationalization of programs.

This tutorial aims to explain plain usage of IOStreams and locales, but also extension techniques for implementing input and output operations for user-defined types as well as new external devices. We explore all the new features in the standard IOStreams library such as the exception mask, sentries, locales, callbacks, etc.

A case study (formatting and parsing of monetary amounts in dual currency environments) runs as a thread through the hands-on labs. It is well suited to demonstrate the use of IOStreams on the one hand, but also use of locales for tackling the culture-dependencies.

## Objective

In this seminar you learn how to use and extend IOStreams and locales effectively.

## Audience

Serious C++ programmers who want to gain an in-depth understanding of the standard IOStreams and locales frameworks.

## Prerequisite

One year of programming experience using C++, or equivalent experience. In-depth experience with IOStreams or internationalization not required.

## Duration

5 days

## Language

English or German

## Course Overview

- 1. USING IOSTREAMS**
  - Formatted Input/Output
  - Stream State and Error Handling
  - File and In-Memory Input/Output
- 2. USING LOCALES**
  - Language-Related Standard Facets
  - Standard Facets for Formatting and Parsing
- 3. EXTENDING IOSTREAMS**
  - User-Defined Inserters and Extractors
  - User-Defined Manipulators
  - lword/Pword & Stream Callbacks
  - User-Defined Streams
- 4. EXTENDING LOCALES**
  - User-Defined Facets
- 5. STREAM BUFFERS**
  - User-Defined Stream Buffers
- 6. CHARACTERS**
  - Characters Encodings, Types, and Traits
  - Code Conversion Facets

## Location

Open enrollment courses are conducted in collaboration with local partner companies. Public courses are offered at regular intervals in

- |                  |                 |
|------------------|-----------------|
| ▫ Amersfoort, NL | ▫ Stuttgart, DE |
| ▫ London, UK     | ▫ Boston, MA    |
| ▫ Zürich, CH     | ▫ Torrance, CA  |

Courses can also be held at your company site. Duration and content will then be tailored to your specific needs and prerequisites.

## Contact

For availability and enrollment send e-mail to [lang@camelot.de](mailto:lang@camelot.de). For further information check out <http://www.langer.camelot.de/Courses.htm>.