Angelika Langer - Training/Consulting

Training by Angelika Langer & Klaus Kreft, 1995 - 2014

Java & C++ Seminars

Course Curriculum

Overview

JAVA

Effective Java	Discusses traps and pitfalls in Java as well as advanced Java features such as inner classes, generics and reflection.
Concurrent Java	In-depth coverage of multithread programming in Java, up-to-date, including the fork.join-framework.
High-Performance Java	Programming for performance: covers programming, monitoring, profiling, and tuning techniques.
Lambdas & Streams	In-depth coverage of programming with lambdas and streams in Java 8.
Java 8	An overview of new language features and JDK extensions in Java 8.
Java Generics	In-depth coverage of programming with generic types and methods in Java.
Java For Beginners	An introduction to Java for programmers without any experience in an object-oriented language.

C++

Reliable C++	Discusses traps and pitfalls in C++, including memory leaks, const-correctness, sequence point problems, and many more. Up-to-date, covers features from C++-11 where relevant.
Effective STL	In-depth coverage of the STL, from introduction over traps & pitfalls to advanced topics such as functors and allocators.
C++ Templates	Covers state-of-the-art template programming techniques, from generic programming to template meta programming.
C++ Exception Handling	Every conceivable aspect of exception handling in C++.
C++ IOStreams	Comprehensive coverage of IOStreams and internationalization in C++, based on our IOStreams book.
C++ For C Programmers	An introduction to C++ for C programmers.

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

3-5 days

Language

English or German

Course Overview

1. CONSTRUCTION AND FINALIZATION
   - Construction & Initialization
   - Destruction vs. Finalization
2. OBJECT INFRASTRUCTURE
   - Object Equality and Comparison
   - Copying Objects (clone())
3. IMMUTABILITY
   - final vs. Constness
   - Dual Class Idiom
4. INNER CLASSES
   - Nested Classes & Non-Static Inner Classes
   - Local & Anonymous Classes
5. LAMBDAS & STREAMS
   - Overview of Lambdas & Method References
   - Overview of Stream API
6. GENERICS
   - Generic Types and Methods
   - Type Erasure and WIldcards
7. REFLECTION & DYNAMIC PROXIES
   - Reflection API
   - Dynamic Proxies
8. WEAK & SOFT REFERENCES
   - Soft References & Caching
   - Weak References & Memory Leaks
9. “NEW” LANGUAGE FEATURES
   - Enum Types, Varargs, Autoboxing Pitfalls
   - Project “Coin” (JDK 7)
   - Default & Static Interface Methods
10. ANNOTATIONS & COMPILER PLUGINS
    - Declaring and Using Annotation Types
    - Compiler-Plugins for Annotation Processing
12. SERIALIZATION
    - Default and Custom Serialization
    - Object Stream Support
13. CLASS LOADING
    - Class Loader Basics
    - Custom Class Loader

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.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

Concurrent Java

- *Multithread Programming in Java*
- *Synchronisation and Thread-Safety Issues*
- *Task and Thread Control*

Course Description

"Writing correct programs is hard; writing correct concurrent programs is harder. There are simply more things that can go wrong in a concurrent program than in a sequential one." (Brian Goetz)

Java supports concurrent programming with multiple threads directly by means of built-in language features. This is one of the many reasons why Java programmers need 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, memory model, etc.

Objective

This seminar aims to give a comprehensive introduction to concurrent programming in Java, exposes students to programming techniques and idioms that have been proven useful in practice, and alerts them to commonly known pitfalls.

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 not strictly required.

Duration

4 days

Language

English or German

Course Overview

1. MULTI-THREADING BASICS
   - Thread Safety
   - Thread Creation
2. CONCURRENCY CONTROL
   - Implicit and Explicit Locks
   - State-Dependent Operations
   - Synchronizers & - Blocking Queues
   - Synchronized vs. Concurrent Collections
   - Thread Local Memory
   - Atomic Scalars
3. THREAD CONTROL
   - Thread
     - States & Priorities
     - Completion, Interruption & Exceptions
   - ThreadPoolExecutor
     - Basics, Setup & Configuration
     - Scheduled Tasks
     - Completion Service & Shutdown
     - Pool Extensions
   - ForkJoinPool
     - Recursive Tasks
     - Principles & Configuration
     - Managed Blocker
4. ADVANCED TOPICS
   - JMM - Java Memory Model
   - Double Checked Locking
   - Atomic Variables
   - Lock-Free Programming

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.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/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

4 days

Language

English or German

Course Overview

1. PERFORMANCE & DEVELOPMENT
   a) Basics
      - Performance Concepts
      - Performance Measuring
   b) Programming
      - Elementary Issues
      - Data Structures & I/O
      - Micro-Benchmarking
   c) Runtime Environment
      - RAM Footprint & Class Loading
      - Garbage Collection
      - JVM Macro-Benchmarks
2. PROFILING, MONITORING & TUNING
   a) JVM Profiling
      - Tool Architecture
      - Profiling Strategies and Tactics
      - Performance Hot Spots
      - Memory Allocation Hot Spots
      - Memory Leaks
   b) GC Profiling (Sun/Oracle)
      - Classic Garbage Collection
      - Garbage First (G1) Collector
      - GC Profiling
      - Throughput Tuning
      - Pause Tuning
      - G1 Tuning
   c) JVM Monitoring

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.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

Java Programming with Lambdas & Streams

- *Java 8 - Language Features & APIs*
- *Lambda Expressions*
- *Parallel Streams*

Course Description

Major extensions to the language and the JDK have been released with Java 8 - most prominently "Lambdas & Streams".

The new language features include lambda expressions and method references, both of which support programming techniques known from functional languages. Now that Java has multiple inheritance via default interface methods programming techniques similar to mixins and traits are possible. In this seminar attendants will gain an overview of all new language features and will practice their use in hand-on labs.

Mastering lambdas is the prerequisite for successfully using streams. Streams are an extension to the JDK collection, which underwent a major overhaul in Java 8. Streams provide a functional API for sequential and parallel bulk operations on sequences of elements. The seminar provides an overview of the stream API from basics such as forEach-filter-map-reduce to advanced operations such as flatMap and collectors. Attendants will practice the use of streams in numerous labs.

The key motivation for leaning about streams is the convenient way in which they support parallelized access to sequences. In order to understand the complex performance model of parallel stream operations the seminar provides insights into the inner workings of streams.

Objective

This seminar aims to give a comprehensive introduction to all new language features in Java 8 including programming techniques and idioms. In addition, the course covers in-depth the new stream API from basics to advanced usage including the complex performance model of parallel streams.

Audience

Software engineers who intend to use parallel streams for performance reason or are generally interested in keeping their Java skills up-to--date.

Prerequisite

Sound knowledge of Java.

Duration

2 days

Language

English or German

Course Overview

1. LAMBDAS
   - Lambdas Expressions
   - Method/Constructor References
   - Functional Interfaces
   - Default & Static Interface Methods
   - Programming with Lambdas
   - Runtime Representation of Lambdas
2. STREAM API
   - Streams & Collections
   - ForEach-Filter-Map-Reduce
   - Fluent Programming
   - Intermediate & Terminal Operations
   - Mappers & Collectors
3. STREAM INTERNALS
   - Pipelining
   - Stateless & Stateful Operations
   - Non-Interference Requirement
   - Sequential & Parallel Execution
   - Performance Model

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.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

New Features in Java 8

- *Java 8 - Language Features & APIs*
- *Lambda Expressions & Parallel Streams*
- *Concurrency Utilities & Date/Time*

Course Description

Major extensions to the language and the JDK have been released with Java 8 - most prominently "Lambdas & Streams".

The new language features include lambda expressions and method references as well as default and static interface methods. In this seminar attendants will gain an overview of all new language features and will practice their use in hand-on labs.

Mastering lambdas is the prerequisite for successfully using streams. Streams are an extension to the JDK collection framework. They provide a functional API for sequential and parallel bulk operations on sequences of elements. The seminar provides an overview of the stream API from basics such as filter-map-reduce to advanced operations such as collectors. Attendants will practice the use of streams in numerous labs.

The key motivation for leaning about streams is the convenient way in which they support parallelized access to sequences. In order to understand the complex performance model of parallel stream operations the seminar provides insights into the inner workings of streams.

The most relevant new concurrency utility is CompletableFuture. It supports asynchronous result processing in a fluent style - yet another API that relies on lambdas. The remaining new concurrency features are for expert users who strive for better performance of their multi-threaded applications.

The seminar also gives an overview of the Date/ Time API, type annotations, and the metaspace.

Objective

This seminar aims to give a comprehensive introduction to all new language features in Java 8 including programming techniques and idioms. In addition, the course covers in-depth the new stream API and provides an overview of further JDK extensions.

Audience

Software engineers who intend to use parallel streams for performance reason or are generally interested in keeping their Java skills up-to--date.

Prerequisite

Sound knowledge of Java.

Duration

3 days

Language

English or German

Course Overview

1. LAMBDAS
   - Lambdas Expressions
   - Method/Constructor References
   - Default & Static Interface Methods
   - Programming with Lambdas
2. STREAM API
   - Streams & Collections
   - ForEach-Filter-Map-Reduce
   - Intermediate & Terminal Operations
   - Mappers & Collectors
3. STREAM INTERNALS
   - Pipelining
   - Sequential & Parallel Execution
   - Performance Model
4. CONCURRENCY UTILITIES
   - Completable Future
   - Stamped Lock
   - Accumulators
   - @Contended
5. MISCELLANEOUS
   - Date/Time API
   - Type Annotations
   - Metaspace vs. PermGen

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.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

Java Generics

- *Language Features*
- *Programming Idioms*
- *Traps and Pitfalls*

Course Description

The 5.0 release of the Java 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, collectively known as "Java Generics". Generics are without doubt the most complex language extension that has ever been added to Java in the past. While the key ideas of parameterized types and methods are easily understood, the devil lies in the detail.

Objective

This course aims to provide a comprehensive overview of the major and minor new features of Java Generics

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

2 days

Language

English or German

Course Overview

Language Features of Java Generics

- *Parameterized Types*
- *Concrete Instantiations*
- *Raw Types*
- *Wildcard Instantiations*
- *Parameterized Methods*
- *Type Parameters*
- *Type Parameter Bounds*
- *Usage*
- *Scope*
- *Static Context*
- *Type Arguments*
- *Wildcards*
- *Wildcard Bounds*

Practicalities - Programming With Java Generics

- Using Parameterized Types and Methods
- Coping With Legacy
- Defining Parameterized Types and Methods
- Designing Generic Methods
- Working With Parameterized Interfaces
- Implementing Infrastructure Methods
- Using Runtime Type Information

Technicalities - Under the Hood of the Compiler

- Compiler Messages
- Type Erasure
- Type System
- Exception Handling
- Static Context
- Type Argument Inference
- Wilcard Capture
- Wildcard Instantiations
- Cast and instanceof
- Overriding of Generic Methods
- Generics and Reflection

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.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

Introduction to Java

- *Language, Tools and APIs*
- *Programming Idioms and Best Practice*
- *Traps and Pitfalls*

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.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

Reliable C++

- *Predictable Program Behavior*
- *Reliable Resource Management*
- *Safe Memory Management*

incl. C++11

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 aims to increase your command of 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 who strive for reliable software**.**

Prerequisite

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

Duration

3 days

Format

Lecture, code reviews, and hands-on exercises. This is not a lecture-only seminar; instead the focus is on gaining practical experience by reviewing and improving real-world source code.

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 & Smart Pointers
   - Move Operations & R-Value References
2. Pointers and Memory Management
   - Correct Allocation and Deallocation
   - 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.

The course schedule can be found at http://www.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

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 adaptors
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.

The course schedule can be found at http://www.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

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, in 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 who care about the runtime performance of their programs.

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.

The course schedule can be found at http://www.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

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 on 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.

The course schedule can be found at http://www.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

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
   - Iword/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.

The course schedule can be found at http://www.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

Introduction To C++

- *All Language Features*
- *OO Programming Principles*
- *Traps and Pitfalls*

Course Description

This course provides an in-depth view of the entire C++ language including templates, exception, RTTI and the STL.

In one week attendants do not only learn all C++ syntax, but also gain experience working with object-oriented programming concepts such as polymorphism and dynamic binding and examine how to use C++ to design and write reusable, maintainable, extensible code. Included are such topics as efficiency tradeoffs, interface design criteria, common design patterns, and other useful idioms and programming techniques.

Audience

Professional C programmers who want to learn C++.

Prerequisite

To fully benefit from this fast-paced and demanding course, you must have significant C programming experience, including working knowledge of pointers, structures, and dynamic memory.

Duration

5 days

Language

English or German

Course Overview

Basic C++ Language Features

- Extensions to C
- Classes
- Inheritance
- Protection
- Constructors
- Dynamic Memory

More Advanced C++ Features

- References
- Inline Functions
- Operator Overloading
- Virtual Functions
- Friends
- Static
- Const
- Names
- Type Conversions

A Deeper Look into the More Complex Features

- Constructors II
- Virtual Functions II
- Inheritance II
- Operator Overloading II

New C++ Language Features

- Templates
- Exceptions
- RTTI
- The Standard Library
- Object-Oriented Programming and Design

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.AngelikaLanger.com/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 contact@AngelikaLanger.com. For further information see http://www.AngelikaLanger.com/Courses.html.

BIO

Angelika Langer

Trainer / Mentor / Author / Consultant

Biography

I work as an independent trainer and consultant with a course curriculum of challenging C++ and Java seminars. My work is backed by 25+ years 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 software development in C++ and Java. I invest a considerable amount of my time on development of explanatory material, including course material, multimedia training, books, and articles.

I conduct lectures, seminars, and workshops worldwide, mainly in Europe and North America. I enjoy speaking at It conferences all over the world. As a Java champion I support the Java Users Groups by joining their meetings and giving presentations. Open enrollment courses are organized at regular intervals, usually in collaboration with partner companies in Germany, Switzerland, and the USA. O-site training is delivered world-wide.

The training business leaves me little time for consulting or 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 Reliable C++, Effective STL, Effective Java, Concurrent Java, High-Performance Java, and Java Generics. For further information see http://www.AngelikaLanger.com/Courses.htm.

Publications

The list of my publications is too long to be mentioned here. For a complete list see http://www.AngelikaLanger.com/Publications.htm.

In brevity: I am author of the “Java Generics FAQ”, an online resource devoted to new language features in Java 5.0. In addition, I am co-author of the authoritative book "Standard C++ IOStreams and Locales" (Addison-Wesley, 2000) and “Java Core Programmierung” (entwickler.Press, 2011). I served as a columnist for C++ Report and C/C++ Users Journal and have been writing the “Effective Java” series (currently published in JavaMagazin) for many years.

Public Lectures

I have been speaking at more conferences world-wide than I can remember. For a complete list see http://www.AngelikaLanger.com/Conferences.htm.

Past lectures were held at JavaOne, OOPLSA, TOOLS, JAX (Germany), OOP (Germany), ROOTS (Norway), Net.ObjectDays (Germany), ECOOP, C++ World (USA), Software Development (USA), DevWeek (UK), ACCU (UK), SDC (Sweden), JDD (Poland), Ch/Open (Switzerland), many JUG meetings, and several other events.

Contact Info

Angelika Langer Training & Consulting
Neumarkter Straße 86d
81673 München, Germany
email: contact@AngelikaLanger.com
URL: http://www.AngelikaLanger.com