Advanced Java 8 using Eclipse (5 Days)

This course teaches how to develop advanced Java applications using Eclipse. The advanced features of Java that developers may be using in many different types of programs are covered. This course also covers many of the advanced features added in Java 7 and Java 8.
This course covers Java SE 8.

What you will learn

After completing this course, the student should be able to:

- Use the logging capabilities built into the Java language
- Work with database data using various Java technologies
- Work with XML from Java code
- Use networking technologies of Java
- Use the support for multiple languages provided by Java
- Utilize the functionality of the Java 8 Date/Time API
- Work with Java Annotations
- Use the security APIs provided with the Java language
- Work with Java NIO and NIO.2
- Work with multi-threaded Java applications
- Work with the advanced Java Concurrency of the Fork/Join Framework
- Define Lambda Expressions
- Utilize the Collections Stream API
- Use the Built-in Lambda Types with Lambda Expressions
- Easily implement many advanced patterns with Lambda functional programming
- Test Java code using JUnit
- Describe the major recent changes in Java

Audience

This course is intended for programmers who are familiar with Java and want to learn about the advanced features of Java.

Prerequisites

A good understanding of object-oriented programming using Java.

Course Outline

Chapter 1 - Java Logging API

- Java Logging API
- Control Flow of Logging
- Logging Levels
- Logging Handlers
- Loggers
- Logging Example
- Logging Formatters & Log Manager
- Logging Configuration File
- Example Logging Configuration File
- Logging Filters

Chapter 2 - JDBC

- Overview
- Types of Drivers
- Making a Connection
- Statements
- Execute, Update and ResultSets
- JDBC and try-with-resources
- JDBC Types
- SQLException
- Prepared Statements
- Stored Procedures
- Transactions
- Connection Pooling

Chapter 3 - Overview of Java Persistence API

- Data Persistence
- Java Persistence API 2.0
- Entities
- Session EJB vs JPA Entities
- Entities
- Persisting and Retrieving Data
- Accessing Entities
- EntityManager & Persistence Unit
- Persistence Context
- Entities - Example
- persistence.xml – Hibernate Provider
- persistence.xml – Open JPA Provider
- persistence.xml - Toplink
- Entity Instance Lifecycle
- Creating EntityManager in Session EJB
- Creating EntityManager in a Plain Java Class
- Working With the EntityManager Interface
- Transaction Basics

Chapter 4 - The Java Architecture for XML Binding (JAXB)

- Introduction to JAXB
- Overview of Data Binding
- JAXB Architecture
- Binding Example
- Binding Framework Example
- Java and XML Mapping Overview
- Namespace and Package Name Mapping
- Simple Type Mapping
- Complex Type Mapping
- Customizing Complex Type Mapping
- Property Accessor Customization
- Property Mapping Customization
- XML Element Mapping
- Mapping Java Enums
- Mapping Collections
- Generating Java Class and Schema
- Marshalling and Unmarshalling

Chapter 5 - Introduction to Lambda Expressions

- Functional Interface
- Anonymous Inner Class (AIC)
- Downside of AIC
- Lambda Expressions
- Lambda Expression Syntax
- Method Reference
- Benefits of Lambda Expressions – An Example
- Initial Version
- Refactor Criteria Into Method
- Predicate Interface
- Using a Predicate
- Implement as Separate Class
- Implement as AIC

- Use Lambda Expressions
- Reuse Lambda Expressions

Chapter 6 - Collections Stream API

Aggregate Operations and Collections

- Pipelines and Streams
- Stream Properties
- Stream API Examples
- Examples Without Stream API
- forEach
- Filtering Streams
- Differences From Iterators

Chapter 7 - Built-in Lambda Types

- Target Types of Lambda Expressions
- Functional Interfaces for Lambda Expressions
- java.util.function Package
- Main Functional Interfaces
- Predicate
- Consumer
- Function
- Supplier
- Primitive Interfaces
- Binary Types
- BinaryOperator and UnaryOperator

Chapter 8 - Collection Operations with Lambda Expressions

- Stream API Review
- Operation Types
- Extracting Data with Map
- Peek
- Search Methods
- Streams are “Lazy”
- Optional Class
- Stream Data and Calculation Methods
- Sorting Streams
- Comparator Interface Changes
- Saving Data From a Stream
- Collectors Class

Chapter 9 - Annotations

- The Annotations Model
- Annotations vs. Descriptors (XML)
- Annotation Types and Annotations
- Built-In Annotations
- JSR 250 Common Annotations
- JSR 250 Common Annotations Example
- Defining Custom Annotations
- Meta-Annotations
- Java 8 Changes – Type Annotations
- Java 8 Changes – Repeating Annotations

Chapter 10 - The Reflection API

- Overview
- Dynamic Facilities In Java
- The Class class
- Examining Classes
- Reflection Allows
- Examining Classes
- Other Things Reflection Can Do
- When Not to Use Reflection

Chapter 11 - Internationalization

- Internationalization
- Adoption Stages
- Internationalization
- Locale
- Dates
- Java 8 – JSR 310 Date-Time Formatting
- User Interface design
- Resource Bundles
- Other Local Customs
- How Java Represents Characters
- Text Files
- Text files

Chapter 12 - Date and Time API

- Working With Date and Time
- Date and Time API
- Date/Time API Packages
- Local Date and Time
- LocalDate
- LocalTime
- LocalDateTime
- Date/Time Methods
- Working With Time Zones and Daylight Saving Time
- Modeling Time Zones
- ZonedDateTime

- Date and Time Amounts
- Calculating Between Days
- Fluent Notation

Chapter 13 - Threads

- Overview of Threads
- Threads in Java Programming
- Write a Runnable Class
- Create Threads
- Another Way of Creating Threads
- Two ways of creating threads
- States in a Thread's Lifetime
- JVM Scheduler
- Control and Schedule Thread
- Coordinating the Concurrency of Multiple threads
- Synchronization
- How Does the Object Lock Flag Work
- Using the synchronized keyword
- The Implication of Synchronization
- Example of Synchronization - Producer/Consumer
- Example of Synchronization – MyStack
- Example of Synchronization – Producer
- Example of Synchronization – Consumer
- Example of Synchronization – SyncTest
- Why Coordination is Required
- Coordinating Thread Cooperation
- wait() and notify()
- Example of Coordination Producer/Consumer
- Example of Coordination – MyStack
- Results
- Deadlock

Chapter 14 - Java Concurrency

- Java Concurrency
- Executor Interface
- Using the Executor
- Callable<V> Interface
- Callable Example
- ExecutorService Interface
- Future<T> Object
- Using Executor, Future and Callable
- Java 8 - CompletionStage
- Java 8 - CompletableFuture
- Java 8 – CompletableFuture Methods
- Atomic Variables
- Using Atomic Variables
- Java 8 – Accumulators and Adders
- Java 8 - ConcurrentHashMap

Chapter 15 - Fork/Join Framework

- Fork/Join Introduction
- Fork Join Tasks
- RecursiveTask
- RecursiveTask Example
- RecursiveAction
- Java 8 - CountedCompleter
- ForkJoinPool
- Java 8 – Common ForkJoinPool

Chapter 16 - Parallel Streams

- Streams Review
- Old-Style Collection Processing
- Stream Collection Processing
- Styles Compared
- Parallel Stream
- Using Parallel Streams
- Pipeline Restrictions
- Avoid Statefulness
- Stateless Implementation
- Deterministic Streams
- Nondeterministic Streams
- Reduction
- Illustrating Parallel Execution
- Performance

Chapter 17 - Networking

- Overview
- URL Connections
- Browser Example
- InetAddress
- Socket Classes
- Simple Clients and Servers
- Simple Clients and Servers (cont.)
- Multithreaded Servers
- UDP Sockets

Chapter 18 - Distributed Objects

- Serialization
- Externalizable
- Remote Method Invocation
- Steps to implement RMI
- RMI – Remote Object Registry
- RMI – Dynamic class loading

Chapter 19 - Java Security

- Overview of JDK Security Features
- Java Cryptography Architecture (JCA)
- Java Cryptography Extension
- Using the MessageDigest Class
- Example of Using the MessageDigest Class
- Using the Signature Class
- Java Security Architecture
- Security Model – Sandbox
- Security Model – Trusted Signed Code & Security Policy
- JDK 1.4 Security Enhancement
- Protection Domains and Security Policies
- ProtectionDomain Class
- Permission Classes
- Using Permission Classes
- Policy Class
- Policy Configuration File
- AccessController Class
- SecurityManager Class
- Using the SecurityManager Class
- Java Authentication and Authorization Service - JAAS
- JAAS – Common Classes
- JAAS - Authentication
- JAAS – Authentication Configuration
- JAAS - Authorization
- Java Security Tools
- Using Java Security Tools – Code Signing

Chapter 20 - Java NIO and NIO.2

- NIO and NIO.2 Overview
- The java.nio.file.Path Interface
- Obtaining a Path Instance
- Path Operations
- Converting Paths
- Operations With Two Paths
- Working With Files
- Java 8 – Files Changes
- File Attributes
- Working With File Attributes
- FileVisitor API
- Finding Files
- Java 8 – Finding Files
- Watching Directories
- WatchService Example
- Buffers
- Channels
- Using Buffers and Channels - Write Example
- Using Buffers and Channels - Read Example
- Working With Legacy java.io.File Code

Chapter 21 - Advanced Functional Programming

- Collections Enhancements
- Map 'merge'
- Creating Streams With Stream.of
- I/O Enhancements
- Files Enhancements
- Flatten Data With flatMap

Chapter 22 - JUnit

- What is JUnit?
- Why JUnit?
- The xUnit Philosophy
- Test-Driven Design
- A JUnit Test
- Running the Tests
- Swing-based Test Runner
- Text-based Test Runner
- JUnit Basics
- assertTrue
- assertEquals
- assertSame
- assertNull
- The Failure Message
- The Test Class
- The Test Method
- The Test Suite

- JUnit with Annotations
- JUnit 4 Test Suite
- JUnit Design
- Testing Strategies
- Specific Techniques
- Testing Simple Java classes
- Testing with Databases
- Testing Web Applications
- Testing Java EE Web Applications
- JUnit with Ant
- Summary
- JUnit with Eclipse
- Create a Test Case
- Test Case "Stubs"
- Running Tests
- Eclipse Test Runner Icons
- Rerun an Individual Test
- Failure Trace
- Debug with JUnit
- Test Suite Wizard

Chapter 23 - Mockito

- The Problem
- Old Solutions
- Bad Solutions?
- What's the other choice?
- Choices
- Mocking stubs
- Mocking Mocks
- Mostly Done
- Other Features
- Spy
- Annotations

Appendix A - Summary of Recent Java Changes

- Java 7 – Major New Features
- Java 7 – Generic Diamond Operator
- Java 7 – Catching Multiple Exceptions
- Java 7 – Rethrowing Exceptions
- Java 7 – try-with-resources Statement
- Java 7 – Suppressed Exceptions in try-with-resources
- Java 7 – Strings in switch Statement
- Java 7 – Changes in Numeric Literals
- Java 7 – Fork & Join Parallel Processing
- Java 7 – NIO.2 File Systems
- Java 8 – Major New Features
- Java 8 – Java Lambda Primer
- Java 8 – Method Reference
- Java 8 – Collections Stream API
- Java 8 – JSR-310 Date and Time
- Java 8 – Default Methods
- Java 8 – Nashorn JavaScript Engine
- Other Miscellaneous Java 8 Changes

Appendix B - Parsing XML with SAX

- SAX
- How it Works
- Core SAX2 Handler Classes
- SAX2 DefaultHandler
- SAX Events
- Ignorable Whitespace
- Parsing a Document
- Using SAXParserFactory
- Parse XML with SAX – Details
- Define an Event Handler
- Create a SAXParserFactory instance
- Define an Event Handler – startElement()
- Define an Event Handler – Element Attributes
- Define an Event Handler – Get Number of Attributes
- Define an Event Handler – Get Name of Attributes
- Define an Event Handler – Get Attribute Values
- Define an Event Handler – An Example
- Define an Event Handler – characters()
- Using characters()
- Define an Event Handler – Error Handling
- Define an Event Handler – ErrorHandler interface
- Parse XML Document
- Simple SAX Parser
- EntityResolver
- Locator

- Document Locator

Appendix C - Parsing XML with DOM

- DOM
- Limitations of SAX
- XML as an Object Model
- Nodes
- The Basic Node Types
- Less Common Node Types
- Node Interface
- Document Interface
- NodeList Interface
- Element Interface
- Attr Interface
- Text Interface
- DOM Parsing
- Parse XML with DOM – Steps
- Prepare DOM Parser Object
- Parse XML Document
- Parse Exceptions
- Example – SimpleDOMParser
- Writing DOM

Lab Exercises

Lab 1. Java Logging
Lab 2. Accessing Database
Lab 3. Using JPA
Lab 4. JAXB Programming
Lab 5. Introduction to Lambda Expressions
Lab 6. Collection Stream API
Lab 7. Using Built-In Lambda Types
Lab 8. Collection Operations
Lab 9. Annotations
Lab 10. Internationalization
Lab 11. Date and Time API
Lab 12. Threading and Synchronization
Lab 13. Java Concurrency Features
Lab 14. Java Fork & Join
Lab 15. Parallel Streams
Lab 16. Networking with Sockets
Lab 17. Distributed Objects with RMI
Lab 18. JAAS Security
Lab 19. Input/Output with NIO and NIO.2
Lab 20. Advanced Functional Programming
Lab 21. JUnit