



Java Advanced - Application Development

Java Training

About The Course

This 3-day Java Advanced course will take you step by step through developing code to use regular expressions to search for strings and utilise streams. Students will learn how lambda expressions work to deal more efficiently with collections, and use patterns to write 'best of breed' applications. You will also learn how to read and write data to files and databases. Most importantly we will cover improving the performance of your code through threads and related technologies, spending time on making sure you understand what thread safety is, and how to avoid the common problems that threads present.

Duration: 3 days

Class size: 10 students max

Times: 9:00am - 5:00pm

Price: Refer to our website for current course and package pricing

After the course?

Each student will receive:

- Certificate of completion
- Training manual
- 12 months FREE email support
- FREE class re-sit (if necessary)

Who Should Do This Course?

This course is designed to students who want to develop complete Java based solutions, with the focus on using built-in functionality that enables you to write multi-threaded code that also interacts with external resources and utilities. It is also ideal for anyone who needs to develop a GUI applications, or is interested in applying patterns to ensure their code follows best practices.

Prerequisites

Participants should have completed the Java for Beginners and Intermediate course or have equivalent skills, including, have a good understanding of Java syntax, coding structure, object oriented programming and familiarity with using IDE's, preferably NetBeans or Eclipse.

Content

Unit 1: Introducing the Course

- About This Course
- Checking Understanding
- Understanding Java as a Platform

Unit 2: Using Regular Expressions

- Introducing Regular Expressions
- Defining Regular Expressions
- Understanding How Regular Expressions Work with Java
- Investigating a Simple Example
- Defining a Pattern

- Understanding the Syntax
- Looking at Examples
- Writing Code
- Working with Pre-Built Patterns
- Looking at Quantifiers
- Looking at Greediness
- Working with Boundary Markers
- Working with Groups
- Looking at Other Useful Methods

Unit 3: Introducing IO

- Understanding a Stream
- Selecting the Right Class
- Selecting a Class Based on Input and Output
- Selecting a Class Based on Stream Content
- Selecting a Class Based on the Type of Destination or Source
- Acting on Streams
- Buffering a Stream
- Filtering a Stream
- Transforming a Stream
- Putting It All Together - Example
- Dealing with Exceptions in Io
- Closing the Stream
- Looking at Typical Exceptions
- Using the Finally Clause
- Putting It Together - Try Catch Example
- Try with Resources
- Understanding Auto-Closable
- Reading and Writing to or from the Console
- Writing to Screen
- Reading from Keyboard
- Understanding Channels
- Investigating Serialisation
- Defining Serialisation
- Understanding the Restrictions
- Understanding How Serialisation Works
- Defining Transient Fields
- Customising Reading
- Customising Writing
- Reading into a Different Object
- Reading into a New Version of an Object
- Using serialVersionUID
- Overriding serialVersionUID

Unit 4: Working with Files

- Introducing the New File Io
- Learning the Advantages
- Finding the Classes
- Avoiding Confusion
- Looking at the New Classes
- Understanding the Path Class
- Understanding the Files Class
- Understanding the Paths Class
- Using the Path and Paths Classes
- Creating a New Path
- Comparing Relative and Absolute Paths
- Using the Basic Methods
- Removing Redundancies
- Defining Sub-Paths
- Resolving Paths
- Relativising Paths
- Dealing with Symbolic Links
- Using the Files Class
- Looking at Available Operations
- Checking for Existence

- Creating Files
- Deleting Files
- Copying Files
- Copying Streams to Files
- Listing Directory Contents
- Using Streams from the New File Io
- Working with File Properties

Unit 5: Understanding Threads

- Introducing Threads
- Understanding What a Thread Is
- Understanding the Advantages of Threads
- Understanding the Disadvantages of Threads
- Looking at a Simple Example
- Making Classes Runnable
- Creating a Thread
- Starting a Thread
- Stopping a Thread
- Understanding the Issues
- Stopping a Thread Explicitly
- Using Interrupt
- Using Sleep
- Understanding Thread Safety
- Looking at a Simple Example
- Looking at the Problems with Shared Data
- Defining Thread Safety
- Looking at Examples of Safe Threads
- Looking at Examples of Unsafe Threads
- Synchronising Methods
- Understanding the Advantages
- Looking at a Code Example
- Understanding Disadvantages
- Synchronising Blocks
- Understanding Advantages
- Looking at a Code Example
- Introducing Locks
- Understanding Deadlock
- Looking at an Example of Potential Deadlock
- Understanding How Deadlock Happens
- Looking at a Simple Solution
- Other Methods of a Thread

Unit 6: Using the Concurrency Package

- Investigating Locks
- Implementing Atomic Operations
- Using Thread Safe Collections
- Introducing Synchronisers
- Investigating Types of Synchronisers
- Understanding the Cyclic Barrier Synchroniser
- Using the Cyclic Barrier Synchroniser
- Using the Executor Service
- Understanding the Executor Service
- Understanding the Runnable Interface
- Returning Results from Callable Tasks
- Understanding the Future Class
- Fetching the Returned Results
- Understanding Callable Tasks

- Shutting Down Executor Services
- Using Fork and Join
- Looking at Code Without Parallelism
- Adding Parallelism to Code
- Breaking the Task Up
- Understanding Work Stealing
- Drawing a Conclusion
- Investigating an Example of Fork and Join
- Looking at a Single Thread Example
- Understanding the Big Picture
- Understanding Recursive Jobs
- Understanding the RecursiveTask Class
- Understanding the RecursiveAction Class
- Splitting Tasks
- Putting It Together

Unit 7: Manipulating Collections Using Streams

- Introducing Pipelines
- Understanding Pipelines
- Understanding the Advantages of Pipelines
- Converting a Collection to a Stream
- Using the Stream to Filter a Collection
- Using the for Each Method of the Stream
- Converting a Stream to a Collection
- Understanding the Nature of a Pipeline
- Revisiting Lambda Expressions
- Saving and Re-Using Lambda Expressions
- Referencing Methods in a Lambda Expression
- Referencing Instance Methods
- Referencing Constructors
- Referencing Static Methods
- Referencing Methods from the API
- Looking at Other Classes in the Streams Package
- Introducing Other Operations on a Stream
- Chaining Methods
- Understanding the Optional Class
- Investigating Intermediate Operations
- Understanding the Filter Operation
- Understanding the Map Operation
- Understanding the Peek Operation
- Understanding the FlatMap Operation
- Investigating an Example of the FlatMap Operation
- Investigating Terminal Operations
- Using ForEach
- Using Collect
- Using Numerical Operations
- Using Other Non-Numerical Operations
- Investigating Short Circuit Terminal Operations
- Looking at Search Operations
- Looking at Lazy Operations
- Drawing Conclusions
- Looking at the Old Way
- Looking at the New Way
- Breaking the New Way into Its Parts
- Comparing the New and Old Ways

Unit 8: Advanced Manipulation of Collections Using Streams

- Sorting
- Looking at Example Output
- Investigating Available Operations
- Using Your Own Comparator
- Using Lambda Expressions
- Adding Additional Sorts
- Reversing the Order
- Putting It All Together
- Collecting
- Looking at Example Output
- Using the Collect Operation
- Investigating the Collector Class
- Averaging
- Grouping
- Joining
- Partitioning
- Putting It All Together
- Parallelising
- Modifying Existing Definitions
- Starting from Scratch
- Understanding Determinism
- Investigating Non-Deterministic Operations
- Looking at an Example of Stateful Operations
- Rewriting Code to Be Stateless
- Using Reduction
- Revisiting Fork and Join
- Understanding Reduction
- Looking at an Example
- Defining Non-Associative Functions
- Understanding Why They Cause a Problem
- Looking at an Example
- Breaking Down the Processing
- Illustrating Parallelism in Streams
- Considering When to Use Reduction

Unit 9: Developing an Application with Java Database Connectivity (JDBC)

- Introducing JDBC
- Understanding JDBC
- Looking at the Classes Used in JDBC
- Understanding the Purpose of Each Class
- Investigating Alternative to JDBC
- Introducing JDBC Drivers
- Understanding JDBC Drivers
- Picking a JDBC Driver
- Locating Your JDBC Driver
- Understanding the URL Syntax
- Investigating Result Sets
- Understanding Result Sets
- Populating a Result Set
- Fetching Data from a Result Set
- Looping Through a Result Set
- Getting Meta Data from Result Set.
- Setting Fetch Size

- Understanding the RowSet Class
- Handling JDBC Exceptions
- Understanding the SQLException Class
- Handling SQL Exceptions
- Closing Resources
- Using Try with Resources
- Writing Other SQL Statements in JDBC
- Using ExecuteUpdate
- Using Execute
- Using Parameters
- Calling Stored Procedures
- Investigating Transactions
- Defining a Transaction
- Understanding Transactions
- Looking at Code Without a Transaction
- Looking at Code with a Transaction
- Understanding Default Behaviour
- Explicitly Controlling Transactions

Unit 10: Patterns

- Introducing Patterns
- Understanding Patterns
- Observing Patterns
- Describing Patterns
- Understanding the Dry Principle
- Understanding Where Patterns Came from
- Getting More Information
- Investigating the Singleton Pattern
- Understanding the Concept
- Using the Pattern
- Coding the Pattern
- Investigating the Factory Pattern
- Understanding the Concept
- Using the Pattern
- Coding the Pattern
- Looking at Advantages
- Investigating the Builder Pattern
- Understanding the Concept
- Using the Pattern
- Coding the Pattern
- Looking at Advantages
- Investigating a More Complex Pattern
- Understanding the Dao Pattern
- Understanding the Problem
- Looking at the Initial Code
- Understanding the Dao Class
- Understanding the Interface and Implementation

- Looking at the Code After Applying the Dao Pattern
- Using the Factory Pattern in Dao
- Looking at the Code After Applying the Factory Pattern
- Investigating Advantages of Using Patterns

Unit 11: Introduction to Swing

- Introducing Swing
- Creating Frames
- Displaying Information
- Working with Fonts, Colours and Shapes
- Displaying Images
- Introducing Layout
- Using Layout Management
- Adding Components
- Understanding Event Handling
- Understanding Actions
- Exploring Mouse Events
- Adding Menus
- Adding Dialog Boxes
- Deploying the Application
- Understanding JAR files
- Understanding Applets
- Using Java Web Start

Unit 12: Introduction to JFX Application

- Introducing Java FX
- Using Scene Builder
- Understanding the Window Structure
- Displaying Text and Images
- Understanding Event Handling
- Laying Out Nodes
- Using Radio Buttons, Mouse Events and Shapes
- Understanding Binders and Listeners
- Using Java FX Collections
- Customising ListView Cells

Looking for course dates?

To view a full list of course dates, please visit our website at www.dynamicwebtraining.com.au

Alternatively please contact our office on **1300 888 724**