

# Advanced Java Course Content

## 1. Generics

- Using Generics
- Type Erasure
- Type Boundaries
- Wildcards
- Generic Methods
- Strengths and Weaknesses of Generics
- Legacy Code and Generics

## 2. The Time API

- A History of Time ... in Java
- Limitations of Date and Calendar
- The Time API
- Temporal Types
- Accessors and Adjusters
- Formatting
- Decomposition Into Fields
- Date Arithmetic
- Managing Precision
- Duration and Period

- Time Zones and Offsets
- Converting Between Time Zones

### **3. The Java Streams Model**

- Delegation-Based Stream Model
- InputStream and OutputStream
- Media-Based Streams
- Filtering Streams
- Readers and Writers
- Byte-Array Streams
- String Readers and Writers
- Closing Streams, Readers and Writers

### **4. Working with Files**

- The File Class
- Modeling Files and Directories
- File Streams
- Working with File Systems
- The Path Interface
- The Paths and Files Utilities
- Processing with `java.util.stream.Streams`

### **5. Delegating Streams**

- Buffering

- Data Streams
- Push-Back Parsing
- Byte-Array Streams and String Readers and Writers

## **6. Java Serialization**

- The Challenge of Object Serialization
- Serialization API
- Serializable Interface
- ObjectInputStream and ObjectOutputStream
- The Serialization Engine
- Transient Fields
- readObject and writeObject
- Externalizable Interface

## **7. Sockets**

- The OSI Reference Model
- Network Protocols
- The Socket Class
- The ServerSocket Class
- Connecting Through URL Objects
- HTTP and Other TCP Servers
- Datagram Clients and Servers
- Non-Blocking Sockets

## **8. Threads**

- Java Thread Model
- Creating and Running Threads
- Manipulating Thread State
- Thread Synchronization
- Synchronized Blocks and Methods
- wait and notify
- join and sleep
- Multi-Threading in Servers

## **9. Concurrency**

- The Concurrency API
- Semaphore and Other Synchronizers
- Concurrent Collections
- Atomic Operations
- Executor and ExecutorService
- Thread Pools
- Parallel Processing

## **10. Reflection**

- Uses for Meta-Data
- The Reflection API
- The Class<T> Class

- The java.lang.reflect Package
- Reading Type Information
- Navigating Inheritance Trees
- Dynamic Instantiation
- Dynamic Invocation
- Reflecting on Generics

## **11. Dynamic Proxies**

- The Proxy Pattern
- Dynamic Proxies in Java
- Use Cases
- The InvocationHandler Interface
- Proxy Classes

## **12. Annotations**

- Aspect-Oriented Programming and Java
- The Annotations Model
- Annotation Types and Annotations
- Built-In Annotations
- Annotations vs. Descriptors (XML)

## **13. Automated Unit Testing with JUnit**

- Automated Testing
- JUnit and Related Tools

- The Test Annotation
- The Assert Class Utility
- Test Runners
- Lifecycle Methods
- Expecting Exceptions
- Test Suites

#### **14. Writing Tests**

- Test Granularity
- Reusing Test Logic
- Recording and Comparing Output
- Test Isolation
- Controlling the Test Environment
- Managing Dependencies
- Non-Invasive Testing
- Designing for Testability
- Factories
- Testing and Threads

#### **15. Test-Driven Development**

- Writing the Test First
- The TDD Cycle
- Advantages of TDD

- Resistance to TDD
- A Case Study

## **16. Mocking**

- Mock Objects in Testing
- Mock Objects in Test-Driven Development
- Static vs. Dynamic Mocks
- Stubbing
- Verifying
- Matching and Capturing
- Using a Spy
- Partial Mocking