# Java SE, Part 3 of 4: JDBC, Reflections, and Threading



**Meet the expert:** Gene Van Sant has over two decades of experience as a software/systems implementer and solutions integration architect. He has experience as a contractor and maintains instructor certifications from IBM and others. Gene balances his career between implementing and automating full blown real world solutions and teaching classes/workshops on emerging technologies and products. This allows Gene to gain valuable extensive real world experience while also staying ahead of the wave with cutting-edge expertise.

**Prerequisites:** This course is intended for programmers who are familiar with Java and want to learn about the advanced features of Java. Students should have a good understanding object-oriented programming using Java.

**Runtime:** 12:48:05

**Course description:** Java was introduced many years ago not for the internet but has since expanded to be an industry standard for web pages. This course goes beyond the introduction to the advanced Java programming language and how to develop Java applications using Eclipse. You will learn how to use JavaBeans, JDBC, Internationalization, Reflection, Threads and Networking.

**Course outline:**

**JavaBeans: Overview**
- Introduction
- Overview: JavaBeans
- JavaBeans Specification
- Why JavaBeans?
- Demo: Simple JavaBean
- Demo: Consume the JavaBean
- Demo: Test the JavaBean
- Summary

**JavaBeans: Structure**
- Introduction
- JavaBean Structure
- JavaBean Life Cycle
- JavaBean Properties
- Demo: Bound Property
- Demo: View JavaBean in GUI
- Demo: Add Events
- Summary

**JavaBeans: Events**
- Introduction
- Events
- Demo: Add/Remove Listeners
- Demo: Add Views in Model
- Demo: Use Views in App
- Demo: Create Wrapper Model
- Demo: Test Wrapper Model
- Demo: Copy Constructor
- Summary

**JDBC: Overview**
- Introduction
- Overview
- Demo: Interface
- Types of Drivers
- Making a Connection
- Demo: Making a Connection
- Demo: Implementing a Connection
- Summary

**JDBC: Statements**
- Introduction
- Statements
- Execute, Update & ResultSets
- SQLException
- Demo: Programming Statements
- Summary

**JDBC: ResultSet & Iterators**
- Introduction
- ResultSet and Iterators
- Demo: Iterators
- Prepared Statements
- Demo: Prepared Statement
- Summary

**JDBC: Stored Procedures**
- Introduction
- Stored Procedures
- Demo: Stored Procedures
- Transactions
- Demo: Transactions
- Summary

**Connection Pooling**
- Introduction
- Connection Pooling
- Demo: Connection Pooling
- Summary

**Localization: Overview**
- Introduction
- Overview:Internationalization
- Locale
- Demo: Built in Locales
- Demo: Default Locale
- Demo: More Locale Information
- Dates
- Demo: Dates
- Summary

**Localization: Resource Bundles**
- Introduction
- User Interface Design
- Resource Bundles
- Demo: Add Resource Bundle
- Other Local Customs
- Demo: Currencies
- Demo: Message Format
- Summary

**Localization: File Data**
- Introduction
- Character Sets
- Text Files
- Demo: Default File Encoding
- Demo: Read/Write UTF-16 Files
- Demo: InputStreamReader
- Demo: OutputStreamWriter
- Demo: Using a Normalizer
- Summary

**Reflection: Overview**
- Introduction
- Dynamic Facilities
- Dynamic Aspect Implications
- The Class class
- Reflection Allows
- When not to use Reflection
- Examining Classes
- Demo: Inspect w/Reflection
- Demo: Inspect Constructors
- Demo: Inspect Fields/Methods
- Demo: Access Super class
- Summary

**Reflection: Objects**
- Introduction
- Manipulating Objects
- Demo: Introspection API
- Demo: Couple Bean w/Model
- Demo: Create Object Reference
- Demo: Use Any Type of Bean
- Demo: Remove Dependicies
- Summary

**Reflection API**
- Introduction
- Creating Objects
- Demo: Reflection API
- Demo: Invoke Class Members
- Demo: Create Instance
- Demo: Interact w/Instances
- Demo: Create Instance w/Args
- Summary

**Reflection: Field Values/Arrays**
- Introduction
- Getting Field Values
- Setting Field Values

*(Continued on page 2)*

# Java SE, Part 3 of 4: JDBC, Reflections, and Threading



- Arrays
- Demo: Invoke Instance w/Args
- Demo: Create List of Types
- Demo: Create UI Calls
- Demo: Arrays
- Summary

**Threads: Overview**
- Introduction
- Overview: Threads
- Why Threads
- Threads in Java
- Two Ways to Create a Thread
- Thread Lifetime
- JVM Scheduler
- Demo: Create a Runnable Class
- Demo: Schedule/Control Threads
- Demo: Change Thread Priority
- Demo: Sleep/Wait
- Demo: Stopping a Thread
- Demo: Create a Daemon Thread
- Summary

**Threads: Synchronization**
- Introduction
- Executor Interface
- Using the Executor
- Callable <V>
- ExecutorService Object
- Future<T> Object
- Demo: Callable <V>
- Demo: ExecutorService
- Coordinate Thread Cooperation
- Synchronization
- Object Lock Flag
- Synchronized keyword
- Summary: Synchronization
- Synchronization: Producer/Consumer
- Demo: Synchronization Producer
- Demo: Synchronization Consumer
- Summary

**Thread: Coordination**
- Introduction
- Coordinate Thread Cooperation
- wait() and notify()
- Coordination: Producer/Consumer
- Demo: Coordination
- Demo: wait() and notify()
- Deadlock
- Demo: Deadlock
- Summary

**Networking: Overview**
- Introduction
- Overview: Networking
- URL Connections
- InetAddress
- Demo: InetAddress Class
- Demo: Simple Browser
- Socket Classes
- Demo: ServerSocket
- Demo: Simple Client
- Summary

**Networking: Multithreaded**
- Introduction
- Multithreaded Servers
- Demo: Multithreaded Servers
- Demo: Multithreaded Client
- Demo: Implement Runnable
- Demo: Add ExecutorService
- Demo: Test Multithreaded Server
- Demo: Object Streams Server
- Demo: Object Streams Client
- Demo: Test Object Streams
- UDP Sockets
- Demo: Datagram Server
- Demo: Datagram Client
- Demo: Test Datagram
- Summary