

# **Java: Under the Hood**

## **Advanced Java**

### **Course Length:**

4 days

### **Audience:**

This course is intended for programmers who have a good working knowledge of the Java programming language.

### **Objectives:**

- Understand object serialization in Java.
- Learn how to create a TCP/IP connection using the java.net package.
- Understand how to use the Reflection API.
- Understand the different types of inner classes in the JDK1.1 and beyond.
- Learn how to create a multithreaded program in Java.
- Understand the lifecycle of a thread and how to synchronize threads.
- Learn how RMI is implemented in Java.
- Understand what JavaBeans are.
- Understand the JavaBeans specification.
- Understand the Java Native Interface.
- Learn how to invoke native code from Java code using JNI.
- Understand the purpose and advantages of JDBC.
- Learn how to connect to a database using the JDBC 2.0 API.

## **Day 1**

### **Advanced Java Topics**

- ❑ **Object Serialization:** Object persistence.
- ❑ **Sockets:** Creating TCP/IP connections.
- ❑ **The Reflection API:** Determining the features of a class.
- ❑ **Inner Classes:** Classes defined within a class or method.

## **Day 2**

### **Threads**

- ❑ **Processes vs. Threads:** Understanding what a thread is.
- ❑ **Thread Scheduling:** The lifecycle of a thread.
- ❑ **Creating a Thread:** The Thread class and the Runnable interface.
- ❑ **Synchronization:** Making your Java classes thread-safe.
- ❑ **The Object Monitor:** Using the wait() and notify() methods.

### **Remote Method Invocation**

- ❑ **Overview of RMI:** Understanding how RMI works.
- ❑ **The RMI Registry:** Stores remote references.
- ❑ **The Remote Interface:** Tagging interface for remote objects.
- ❑ **Implementing RMI:** The steps involved in implementing RMI.
- ❑ **Callbacks:** Passing a remote reference.
- ❑ **Activatable Objects:** Instantiating remote objects on demand.
- ❑ **The Activatable Class:** For use with activation.
- ❑ **The Setup Class:** Registering the remote interface with rmid.
- ❑ **Implementing Activation:** The steps involved in using rmid.

## **Day 3**

### **Java Beans**

- ❑ **Overview of Java Beans:** Java software components.
- ❑ **The Bean Development Kit:** The bean specification and Sun's beanbox.
- ❑ **Properties:** Adding properties to a bean.
- ❑ **Methods:** A bean's behavior.
- ❑ **Events:** How beans communicate with each other.

## **Day 4**

### **Java Native Interface**

- ❑ **Overview of JNI:** Interface to native code.
- ❑ **Using JNI:** The steps involved in using JNI.
- ❑ **“Hello, JNI”:** Your first JNI application.
- ❑ **JNI Data Types:** How the Java data types map to JNI’s data types.
- ❑ **The Methods of JNI:** A reference for the JNI methods.
- ❑ **Fields, Methods and Constructors:** Accessing members of Java objects from native code.

### **JDBC**

- ❑ **Overview of JDBC:** An API for communicating with databases.
- ❑ **JDBC Drivers:** Understanding the various types of drivers available for JDBC.
- ❑ **Connecting to a Database:** Loading the appropriate driver and making the connection.
- ❑ **Statements:** Executing SQL statements.
- ❑ **Prepared Statements:** Executing precompiled SQL statements.
- ❑ **Results:** Working with result sets.