

TT6600 Architecting and Designing J2EE Applications (5 days)

Geared for software architects and designers using J2EE in developing n-tiered systems, *Architecting and Designing J2EE Applications* is a J2EE architecture training course that provides students with knowledge needed to use the Java 2 Platform, Enterprise Edition (J2EE platform) to create robust enterprise applications that allow for rapid change and growth.

COURSE SNAPSHOT

Duration: 5 days (can be tailored)

Focus: J2EE applications – J2EE architecture and design

Targeted Audience: Experienced Java / J2EE Developers; Architects & Designers

Skill-level: Intermediate

Hands-On Format: Approx 50/50 lab to lecture: Extensive hands-on programming labs; Expert lecture combined with open discussions and in-depth demonstrations

Language / Tools: Java 5 or Java 6 delivered with JBoss™ 4 and Eclipse / Ganymede. Please inquire about other software options.

Delivery Format: Available for onsite private classroom presentation, or live online / virtual presentation

Customizable: Yes – can be blended with other courses / topics, tailored to cover specific areas of interest, or tailored to include your organizations specific challenges / examples

Participants will gain an understanding of the strategies needed to create application blueprints that work well when implementing J2EE technologies. These strategies include effective decision making through the use of systemic qualities (such as scalability and flexibility), J2EE technology blueprints and design patterns, and iterative and incremental development.

► Course Objectives: What You'll Learn

Students who attend **Architecting and Designing J2EE Applications** will leave the course armed with the required skills to design, implement, and deploy robust, secure web applications. The course begins with a high-level look at J2EE from a Service Oriented Architecture and Component-Based Design perspective. Students then use a CBD process on a case study to design a solution that is then translated to J2EE and its components. The J2EE technical architecture is used to implement the solution.

Working in a dynamic, interactive discussion and hands-on programming environment, let by our J2EE expert practitioners, students will:

- Understand the concepts of Service Oriented Architectures and Component-Based Design and how they relate to J2EE
- Apply CBD requirements gathering and design process to real-world problems
- Relate a design to the J2EE technical architecture
- Recognize and use the various tiers that are supported in J2EE
- Understand and use basic design patterns within J2EE applications and architectures

The course provides a solid foundation in essential terminology and concepts, extended and built upon throughout the engagement. Processes and best practices are discussed and illustrated through both discussions and group activities.

Attending students will be led through a series of advanced topics comprised of integrated lectures, extensive hands-on lab exercises, group discussions and comprehensive demonstrations. Please see below for additional information about the hands-on lab work.

► Audience & Pre-requisites: Who Should Attend

This is an **intermediate to advanced-level** J2EE training course, geared for designers and architects that need to relate real problems to J2EE-based solutions. Attendees should have practical experience developing basic J2EE applications.

► Related Courses – Suggested Learning Path

Take Before: Students should have basic development skills and a working knowledge in the following topics, or attend these courses as a pre-requisite:

- **TT4000 Understanding Internet Architectures**
- **TT2100 Core Java Programming Fundamentals**
- **TT5320 Developing JEE Component Compliant Applications or TT5100-JEE JEE Web Application Fundamentals**

Take Instead: We offer other courses that provide different levels of knowledge or focus:

- JEE developers may consider **TT6200 Architecting and Designing Service-Oriented JEE Applications**

- For the treatment of J2EE patterns, consider **TT6500 J2EE Design Patterns**

Take After: We offer a variety of introductory through advanced security, development, project management, engineering, architecture and design courses. Students may want to consider the following topics as follow-on options for this course:

- Java Security or JEE Security topics
- Java EE / JEE topics: EJB3.0; Spring; Hibernate; Design Patterns & more – Introductory through advanced level
- Service-Oriented Analysis and Design
- Web Services – Intro through Advanced
- AJAX, XML or other Web Development topics
- Architecture & Analysis courses
- Software engineering or business analysis courses

Please note all development courses may also be offered in other programming languages or tailored to suit your unique requirements. We will work with you to come up with the best solution to ensure your needs are met, whether we customize the material, or devise a different educational path to help your team best prepare for this training. Please contact us for recommended next steps tailored to your longer term education, project or development objectives.

► **Experiential Learning: Hands-On Labs**

This class is “technology-centric”, designed to train attendees in essential JEE development skills, coupling the most current, effective techniques with the soundest industry practices.

This workshop is about **50% dynamic lab exercises** and **50% lecture**. Throughout the course students will be led through a series of progressively advanced topics, where each topic consists of lecture, group discussion, comprehensive hands-on lab exercises, and lab review. Multiple detailed lab exercises are laced throughout the course, designed to reinforce fundamental skills and concepts learned in the lessons. At the end of each lesson, developers will be tested with a set of review questions to ensure that he/she has fully understands that topic.

► **Delivery Environment: Tools to Use**

Although this training is skills-centric, this course can be delivered using a variety of JEE compliant software combinations, including but not limited to: Eclipse / Ganymede, MyEclipse, IBM® WebSphere Rational Application Developer (RAD7), Oracle JDeveloper or other IDEs. Application server options include IBM WebSphere, Oracle WebLogic, Apache TomCat, JBoss and others. Please inquire for details and options.

Our detailed lab workbooks are complete with software-specific instructions, screen shots and step-by-step tutorials for using the software you select. In most cases, we can easily port our classes to run in the environment of your choosing.

► **Easy & Secure Set Up! LoadNGo™ Instant Classroom Kit**

For course deliveries or virtual presentation using open-source tools, we'll provide our unique **LoadNGo Instant Classroom Kit**, which enables students to run the entire course off of a DVD that hosts the entire course set up software, labs, and other pertinent useful educational resources, whitepapers and more. You only need to provide the hardware and appropriate O/S, and we'll do the rest. No installation needed. **Great for secure environments.** Minimum set up burden for your team or firm, with maximum results for your students.

No matter which set up option or software your firm requires, we're pleased to provide a detailed set up guide for all private or on-site courses, and as much assistance as you require to prepare your students or classroom for the course. Our support personnel and instructors can be contacted for any advice you may require to prepare your classroom and/or students for attendance.

► **Student Materials: What You'll Receive**

Our robust course materials include much more than a simple slideshow presentation handout. Student materials include a comprehensive hard-copy course manual, complete with detailed course notes, code samples, diagrams and current reference materials, all directly related to the course at hand, indexed for ease of use. Step-by-step lab instructions and project descriptions are clearly illustrated and commented for maximum learning.

In addition to everything students need for the course, the **LoadNGo Instant Classroom Kit** described above also includes of workshop labs and solutions; non-restricted workshop software, APIs, documentation, technical education papers, and specifications and tutorials pertinent to the training course. Our course kits are designed to serve as an excellent and useful reference set, long after we leave your classroom.

► **Optional Pre / Post-Testing & Skills Assessment**

We work with you to ensure that your resources are well spent. Through our basic course pre-testing and/or post-course assessments, we ensure your team is up to the challenges that this course offers. Our goal is to structure the best solution to ensure your needs are met, whether we customize the material, or devise a different educational path to prepare for this course.

Please contact us for details about our online pre and post test assessment services, custom managed training plans for one student or your entire organization, or our custom online training program management system for monitoring the courses or progress while skilling your students of all experience levels.

► **Bridging the Gap: Collaborative Mentoring Services**

Our team of technical experts is also available for various project assistance services to help your team apply their newly-learned classroom skills to their real-world project in a meaningful, practical way, right after the training ends.

Our custom **collaborative mentoring programs** integrate with or extend your team's classroom training experience, to help bring these skills into existing (or inherited) legacy projects, into new projects, or to simply keep your students sharp them in between projects. Our programs can be highly involved and closely integrated with your project timelines or group development

efforts, or can be less involved, serving simply as an overarching educational framework or 'spot check' to keep your group skills moving forward in between projects or waiting for projects to begin. Please contact us for details about this exciting custom service.

Workshop Topics Covered

Need more info? Please note that a more detailed outline of the course table of contents, lists of lab exercises and project descriptions is available. Please contact us at Training@triveratech.com for info.

Need courseware? This course is fully customizable, and also available for license with complete support for qualified organizations. Please contact Courseware@triveratech.com for details.

Session: The Business Architecture: Service and Component Based Design/Development (sCBD)

Lesson: Introduction to Service Orientation and CBD

- Understand the move in IT which leads us to Components and the direct business value it can bring.
- Understand the possible effect of Components on IT budgets.
- Define a Component
- See how a Component is rendered in UML2.0 and its relationship with a Port.
- Define Component Based Development and Design.
- Understand the relationship between Enterprise Architecture and (the need for) Components.
- Understand the relation between SOA and CBD
- Understand how the functional requirements find their way through the Business Architecture.
- Describe the requirements of the Software Development Process that leads to components.
- Understand the potential problem of a Waterfall approach.
- Define iterative and incremental development.

Lesson: Describing a Component Based Software Development Process

- Examine the gap in Software

- Development between the model and the domain experts (the source).
- Examine the gap in Software Development between the model and the developers (the goal).
- recognize the area of tension between these two gaps.
- Recap UML (based on version UML2 including relation between UML, MOF and XMI)
- Understand that (traditional) OO Development does not/might not leverage all the possibilities of SOA.
- Understand the concept of "Design by Contract"
- Describe the requirements of the Software Development Process that leads to services and components.
- Understand the potential problem of a Waterfall approach.
- Define iterative and incremental development.

Session: Applying sCBD's Component Modeling

Lesson: The Requirements Capture Phase

- Use-Cases are covered to get a complete picture of a Software Development process that delivers Components for your technical environment.
- Understand how to gather system requirements.
- Introduce Use-Cases for the other lessons on component and solution delivery.
- Describe a way of documenting non-functional requirements

Lesson: The Business Architecture / Component Design phase

- Understand the steps involved when specifying the component architecture.
- Illustrate a top-down approach for identifying components.
- Describe an informal process for identifying components.

- Describe a formal process for identifying components.
- Explain how you can guarantee that you have all services in place.
- Understand how to model document-oriented services

Session: Enterprise Architectures and a Justification for using J2EE

Lesson: The Technical Enterprise Architecture

- Define the Enterprise (Technical) Architecture and its relationship to Project Architectures.
- List the four technical characteristics/requirements of a Technical Enterprise Architecture.
- Understand how the non-functional requirements find their way through the Technical Architecture.
- For each of the four requirements list possible solutions.
- For each of these solutions, list possible technical approaches.
- Understand the challenges involved with these approaches.
- Understand the nature and reason for the technical challenges that were created.
- Explain what Middleware can address and what other challenges it brings to the table.
- Explain the need for a Framework.
- Introduce two Frameworks: .Net and J2EE

Lesson: J2EE as a Candidate Framework for your Architecture

- Get a brief introduction of the J2EE Framework within this context.
- Recap the J2EE Framework architecture.
- Recap the Technical Architecture challenges and recap the technical solutions.
- For each of the technical solutions previously described, understand how J2EE can address these (or help

you).

- Main goal of this part is to determine a foundation for choosing J2EE as the technical framework for your project or EA.
- Overview of the complete architecture briefly touching all tiers and their associated J2EE and J2SE technologies.

Lesson: J2EE Vendors And Market

- List some of the important players in the J2EE Application Server region
- Through the list of players, describe the different types of Servers
- Discuss criteria for choosing an Application Server

Session: Technical Overview of J2EE

Lesson: Overview of the Complete Architecture

- Get a complete technical overview of the J2EE Platform
- Recap how each tier is addressed in J2EE
- Recap the architectural role of each of the technologies
- Recap the concept of the EAR and the deployment descriptors
- Understand / recap the relationship between J2EE and Web Services
- List the supporting technologies (JMX, JACC, JTA, etc.)
- List the different versions of J2EE and the related versions and APIs

Lesson: (J2EE) Design Patterns Summary

- J2EE Business Tier Pattern: Business Delegate (Proxy)
- J2EE Business Tier Pattern: Value Objects (Holder).
- J2EE Business Tier Pattern: Service Locator.
- J2EE Integration Patterns: Data Access Objects.
- J2EE Integration Patterns: Service Activator.
- Understand how motivations of certain J2EE Patterns have been addressed by a top-down approach of CBD.

Lesson: Presentation Tier

- List the architectural responsibilities of a Servlet and a JSP
- Understand the role of the Web

Container

- Recap Servlet implementation
- Recap JSP implementation
- Understand the architectural role of Custom Tags
- Recap Custom Tag development and use
- Understand the architectural position of JSF
- Recap an Web Application and the web.xml

Lesson: Application Tier

- Understand the three different ways of implementing the application tier.
- Understand Enterprise JavaBeans
- Understand the way an EJB container works
- List the different types of EJBs
- Recap EJB development
- Understand different ways of exposing services as a Web Service
- List the contents of the deployment descriptor
- Understand the life cycle of a Session and an Entity Bean
- Understand Transactions within the context of a distributed environment
- Understand the importance of the transaction isolation level
- List criteria for choosing the transactional configuration for your application(s)
- Understand the relationship between EJB, IIOP, SOAP and GIOP

Lesson: EIS Tier

- Understand the different technologies to connect with EIS
- Understand the difference between access and integration
- List requirements for integration
- Recap JDBC, JMS and the Connector Architecture

Lesson: Security

- Understand the security tasks and how they are addressed in J2EE
- Understand how to establish proof of identity in J2EE (web and desktop clients)
- Understand how to restrict access to parts of the web application
- Understand how to restrict access to services
- Understand how to establish data integrity and privacy

- Understand that SSL might not be sufficient in B2B web services
- Explain the configuration of security and the power of indirection
- Discuss different security configurations (including propagation and delegation)

Session: Mapping to the Technical Architecture

Lesson: Mapping Components to a (distributed) J2EE Environment

- Understand how the Business Components map to J2SE Components
- Understand how the Business Components map to J2EE Components.
- Understand how this process maps to MDA
- Make a well-founded decision when to use EJBs (or not).
- Emphasize once more the importance of non-functional requirements.
- Understand the impact of a Distributed Architecture on the Business Architecture (i.e., signatures of services)
- Appreciate the Value Object Pattern (including JavaBean, Structure and XML strategies).
- Discuss how Technical Components (e.g., Data Access) might be separate components or sub-components.
- Understand the difference of data transfer between local and a distributed Component Model
- Choose a deployment topology based on QoS requirements

Lesson: The Internal Design Phase (Component Delivery)

- Understand the steps involved in the internal design of components
- Discuss patterns and best practices can be used inside the component design.
- Understand the role of the EJB local interface in component design.
- Understand how parts of XP can be used in this phase

Lesson: Overall (Web-Based) J2EE Architectures

- Discuss different J2EE Web

- Application Architectures.
- Understand the benefits and challenges for each of these architectures.
- (Re)establish the architectural role of JSPs, Servlets, Custom Tags, EJBs, etc.
- Possible role in the Presentation tier of the XML/XSL combination in J2EE Applications

Session: Path to Useful Web Services

Lesson: Services via the Web

- Service Defined
- SOA Defined
- Organizational Framework
- Technical Framework
- Orchestration
- Services vs SOA
- SOA in the past
- What is new in SOA
- Business impact/ROI
- Myths/Reality
- Adoption issues
- Characteristics of a Good Service
- Services and their Formal Contracts
- Services Should be Stateless
- Service Design Guidelines
- SOA Anti-Patterns

Lesson: Web Services Overview

- Crossing Boundaries
- What are Web Services?
- Six Key Components
- Web Services Characteristics
- Web Services Architecturally
- Technology Comparison
- Architectural Perspective
- Web Services Enable Decoupling
- Many Web Services Challenges
- Secure Services
- Spec and Standard Evolution
- Web Services Interoperability Organization
- WS-I Has Many Deliverables
- Basic Profile 1.0 Consists of:
- Has > 100 Requirements and Suggestions
- .NET Platform & .NET Web Services
- Java and Web Services

Lesson: Web Services, Java, and J2EE

- XML and Java APIs at a Glance
- XML Signature
- XML Digital Signatures
- XML Encryption

- JAXP
- JAXB
- JAX-WS
- SAAJ
- JAX-WSA and XWSS
- Web Services APIs
- Web Services for J2EE (JSR109)
- J2EE and Web Services
- Web Services Metadata
- Web Services Stacks at a Glance
- WSIT
- Apache Axis2
- JBossWS
- JWSDP
- WebSphere WS
- Spring-WS
- Key Features

Lesson: Web Services Quickstart

- What is WSIT?
- WSIT Tool Support
- How is WSIT Used?
- Web Service Development with WSIT
- Debugging Web Services
- TCP/IP Monitors Provide View of Wire

Session: Foundation - XML and Java

Lesson : XML, Namespaces, & Schema

- What is XML?
- XML Can Provide Application-Specific Information
- Content: XML Document Syntax Rules
- Structure: A Document Type Definition
- XML Transformation to HTML
- XML Separates Structure, Content and Format
- TriveraTunes Purchase Order
- Content as Markup
- Tell Parser That Text is Data
- Use Predefined Entities
- Well-Formed and Valid XML Documents
- Why Are These Definitions Important?
- XML Namespaces
- Name Collision – Example
- Inter-Organization Name Collisions
- W3C's Solution: Namespaces
- Uniform Resource Indicator
- Declaring a Namespace
- Namespace Scope
- Parsers Use URI, Not the Alias

- Default Namespace
- Attributes and Namespaces
- Example of Namespaces
- Namespaces Best Practices
- Benefits From Valid XML
- W3C XML Schemas
- Impacts of Schemas
- General Form of an XML Schema
- Elements, Attributes, and Types
- Simple Schema and XML Document
- Element Definitions
- Corresponding XML Schema
- Simple Types - Primitive Datatypes
- Restricting Simple Types: Facets
- Complex Types Bring More to Validation
- Repetition Control
- Restricting Simple Types
- Complex Types Can be Derived
- Derivation by Extension
- Extension of Phone Number
- Associating Schemas with XML Instances
- Using XML Schema with Namespaces
- Namespaces Provide Thread of Connection
- Schema Defines a Target Namespace
- XML Doc Uses Schema-Defined Namespace
- schemaLocation Links Namespace to Location
- Relating Schemas to XML

Lesson: XML in Java - JAXP and JAXB

- XML Parsers Are Complex and Powerful
- Parsers Are Integral to XML Processing
- Parsers and API's
- Parser Generates DOM, Then Hands to App
- Parsing With a DTD or Schema
- Many Options to Consider
- XML and Java
- Security Concerns Relative to Parsing
- Bridging Application Data and XML
- JAXP: Java API for XML Processing
- JAXP and Transformations
- Challenges to Mapping XML
- Generating XML is Nondeterministic
- JAXB: Binding XML to Java
- JAXB 2.0 Incorporated Changes
- JAXB 2.0 and Java Versions
- Defining the Rules in JAXB
- Turning Rules into Java Classes
- Using the Generated Classes

- Creating Content
 - Some JAXB Type Bindings
 - XML Schema for List of Items
 - Corresponding Class Interface
 - A Word About Validation...
- Hands-On Labs & Tutorials**
- Define The System Requirements (Creating the PIM)
 - Identifying the Business Components (Elaborate the PIM)
 - Identify Component Services
 - Writing a Servlet and a JSP applying MVC (Optional)
 - Writing a Session Bean (Optional)
 - Define High-Level Technical Architecture for the Business Tier
 - Mapping the Component Model to J2EE (Creating the PSM)
 - Design Components (“Design by Contract”)
 - Web Services in Action
 - Implementing a Web Service
 - Debugging a Web Service
 - Namespaces and Schemas
 - Working with JAXB
 - Tutorial: Configuring JBoss Using MyEclipse
 - Tutorial: Creating J2EE Projects in MyEclipse
 - Tutorial: Deploying J2EE Projects into JBoss
 - Tutorial: Configuring the Database in MyEclipse

► Why Work With Trivera Technologies?

Whether you are a project leader choosing a training provider or course to bring to your team, or an organization or an instructor looking to potentially license or use course materials to train your own team, or a student looking for an exciting, targeted training class to attend or to recommend to your colleagues - ***Our single focus is to make YOUR training event or experience a success.*** Here’s why choosing Trivera Technologies as your J2EE JEE training & courseware resource takes the risk right out of your decision making process...

- **We provide a solid J2EE / JEE application development foundation.** Students will learn how to develop (and reuse!) essential JEE development and design skills and concepts properly, using best design practices, grounding them for advanced curriculum. Students will be prepared for designing and implementing real solutions, right after the class ends. Students will learn the importance of developing well-designed enterprise Java applications.
- **Our courses are focused - no "fluff" included.** We offer more than a “laundry list” approach to teaching. All lessons have clear objectives, are fundamental to core Java, JEE or J2EE development and design practices, and are reinforced by hands-on labs and solid practical examples. Each lesson has performance driven objectives that ensure students will learn technologies and skills core to fundamental server-side application design – nothing more, nothing less.
- **Our materials are comprehensive, and current.** Our comprehensive manuals include not only a hard copy of the course presentation, but also detailed reference notes, pertinent diagrams and charts, current lists of suggested online resources and articles, and often technical tutorials or white papers geared to the topics at hand. Our dedicated course development team keeps everything as current as possible with both industry trends and software editions to ensure your team is getting the most current information available.
- **We set you up!** Hands-on courses also include our unique materials for each student, complete with our **LoadNGo Instant Classroom** course set up DVD, software, and a multitude of learning resources that complement the course. Run the course right off the DVD – minimal set up for your company – maximum results for your students.
- **We foster "Learning by Doing".** Progressive labs are designed in such a way that students get a firm grasp on fundamental skills while they work toward designing a complete application. All labs are take-home, and all solution code is presented in an easy to use self-study format for future use and review.
- **We have to adhere to higher standards.** As a courseware provider to other organizations, training firms or independent instructors, our content and hands-on lab materials are licensed internationally by dozens of firms, and are therefore subject to very stringent quality requirements. Not only will your organization benefit from our own technical team’s technical expertise, but also the feedback of hundreds of students and trainers using these materials, worldwide, on a regular basis. This unique fact guarantees that our materials are not only robust and interesting, but also technically correct, current and of the highest quality and usability.
- **We bring years of practical, current experience into the classroom and content.** Our instructors and course authors are also skilled mentors, Java, JEE/JavaEE, J2EE, .Net, Agile, SOA, and web services developers, architects and security-oriented professionals. We believe that learning, using and maintaining solid software execution and delivery methods are as important as gaining sharp coding skills. Best Practices for software development and execution, beyond technical coding skills, are enforced throughout all of our courses and discussions. Our team brings this extensive experience into every classroom and engagement. Our team has trained thousands of students.
- **We're skills-centric.** Although our team has extensive experience using a variety of tools and solutions, our core content is “technology-centric”. Our aim is to teach you the best skills and solutions out there – not to sell you software from any particular vendor.
- **We're Java & JEE / J2EE authors and industry speakers.** Our team was selected to write the online *J2EE, EJB, EJB CMP-CMR and Web Services Tutorial Series for IBM developerWorks®* (www.ibm.com) These are the same instructors who train our classes and author the courseware. Most of our trainers/consultants have also authored additional articles on web services, EJB, Struts, J2EE / JEE and advanced Java topics, and are recognized speakers and presenters on the industry technical seminar circuit. Our team is comprised on several successful published authors. Members of our team have written or contributed to: *Eclipse Kick Start, Mastering Eclipse; Professional Jakarta Struts; Using Java Tools for Extreme Programming; Mastering Resin; Mastering TomCat and others.*

- **Our services are guaranteed.** Whether you're a stakeholder organizing your firm's educational services, a student in our live or virtual classroom or a trainer using our materials to educate your own client or team – **Our core mission is to make YOU a success in the classroom.**

► **For Additional Information**

Need dedicated training? All courses can be brought onsite or run virtually for a **private presentation**, anywhere around the world, customized to suit your unique requirements or goals. Please contact Training@triveratech.com for course details, Public Schedule dates and locations, and Special Discount Offers.



*Trivera Technologies is a 100% Female-Owned Small Business Concern
GSA Schedule # GS-35F-0188T
Please contact us for details & pricing.*

Need courseware? **Let us take the risk out of your classroom delivery!** All materials are also available on a worldwide basis for corporate license with complete instructor support and free corporate branding. Our LoadNGo Set up is available to partners as well! We guarantee our pricing and service. Samples of our course materials, as well as live client references for all of our services are available upon request. Please contact Courseware@triveratech.com for details.

For more information about our training, mentoring or courseware development or licensing options, or to see our complete list of course offerings and services, please visit us at www.triveratech.com, email Training@triveratech.com or call 609.953.1515.