

Advanced Java Using Galileo Eclipse

Course Summary

Description

This course is designed for programmers who are familiar with Java and who want to learn about the advanced features of Java and server side programming.

Topics

- Using Eclipse and Tomcat Server
- Servlets
- Overview of HTML
- Java Server Pages
- JavaBeans
- JDBC
- Servlets and JSP Interaction
- Configuring Deployment Descriptor
- Security
- Threads
- Networking
- Distributed Objects with RMI
- Overview of Enterprise JavaBeans (optional)
- Overview of J2EE Architecture (optional)

Audience

This course is designed for programmers who are familiar with basic java programming.

Prerequisites

The student should have a working knowledge of Java programming and some experience with HTML.

Duration

Five days

Advanced Java Using Galileo Eclipse

Course Outline

I. Getting Started with Eclipse and Tomcat

- A. Software Requirements
- B. Eclipse
 - 1. Platform
 - 2. Architecture
 - 3. Getting Eclipse
- C. Eclipse
 - 1. Architecture
- D. Apache Tomcat Server 5.0
- E. J2EE and Tomcat
- F. Application Server Connection

II. Servlets

- A. What is a Servlet?
- B. Typical Uses of Servlets
- C. Advantages Over CGI
- D. How Servlets Work
- E. Java Servlet Architecture
- F. Servlet's Lifecycle
- G. The service() Method
- H. Writing MyServlet
 - 1. Creating HelloServlet in Eclipse
 - 2. Application Connector
 - 3. Running MyServlet in Eclipse
 - 4. Deploying the Web Project
 - 5. Start Apache Tomcat
 - 6. Invoke HelloServlet
 - 7. Deployed mySample in Tomcat Server
 - 8. Deployment Descriptor
- I. The ServletRequest Object
- J. Handling Form Data
- K. doGet() vs doPost()
 - 1. Using doGet()
 - 2. Using doPost()
- L. The ServletResponse Object
- M. Session management

III. Overview of HTML

- A. HTML
- B. Data Structure
 - 1. HTML tags
 - 2. Tag Syntax
 - 3. Types of Tags
- C. HTML Form Tags
- D. Page Designer
- E. Creating HTML files in Eclipse

IV. JavaServer Pages

- A. JavaServer Pages
- B. Why use JSP?
- C. How JSPs Work
- D. JSPs ILife
- E. Writing HelloWorld.jsp
 - 1. Creating HelloWorld.jsp in Eclipse
 - 2. running HelloWorld.jsp in Eclipse
 - 3. Deploying the Web Project
 - 4. Start Apache tomcat
 - 5. Invoke HelloWorld.jsp
 - 6. Deployed mySample Web Project in Tomcat Server
- F. Web Project Directory Structure
- G. JSP Tags
 - 1. Directives
 - 2. Action
 - 3. Scripting elements
 - 4. Comment
- H. The page directive
- I. Scripting elements
 - 1. Scriptlets
 - 2. Expression
 - 3. Declaration
 - 4. Comments
 - 5. Using JSPs Scripting Elements
- J. Implicit Objects in JSP
 - 1. Inserting Implicit Objects in Eclipse

Advanced Java Using Galileo Eclipse

Course Outline (cont'd)

V. JavaBeans

- A. JavaBeans
 - 1. JavaBean Structure
 - 2. Properties
 - 3. A Simple Example
- B. Standard action
 - 1. <jsp:useBean>
 - 2. <jsp:getProperty>
 - 3. <jsp:setProperty>
- C. JavaBeans in JSPs
- D. Importing files in Eclipse
- E. Generate setter and getter methods

VI. Integrating Servlets and JSPs

- A. Calling a servlet from JSP
 - 1. Using the FORM tag
 - 2. <jsp:include>
 - 3. <jsp:forward>
 - 4. Adding <jsp:include>
 - 5. Adding <jsp:forward>
- B. Call a JSP from a Servlet
 - 1. Using request dispatcher
 - 2. Using the sendRedirect() method of the response object
- C. SendRedirect() vs forward()

VII. JDBC

- A. What is JDBC
- B. Software Requirements
- C. JDBC Driver Types
- D. Basic JDBC interaction
 - 1. Load the driver
 - 2. Establish a connection
- E. Execute SQL statements
- F. Statement
 - 1. execute()
 - 2. executeQuery()
 - 3. executeUpdate()
 - 4. ResultSet
 - 5. Inserting a record
 - 6. Updating a record
 - 7. Deleting a record
- G. Prepared statement
- H. Callable statement
- I. DataSource
 - 1. Using a DataSource

- 2. Configure JNDI DataSource in Tomcat
- 3. Define DataSource Reference
- 4. Using the DataSource

VIII. Configuring Servlets Using web.xml

- A. Deployment Descriptor
- B. web.xml
- C. Deployment Descriptor Diagram
- D. web-app Element Structure
 - 1. <servlet> Element Structure
 - 2. Adding a Servlet in web.xml
 - 3. <servlet> element for servlet
 - 4. <servlet> element for JSP
 - 5. <servlet-mapping> element
 - 6. Custom URL for a servlet
 - 7. Custom URL for a JSP
- E. Initializing Parameters
- F. Adding Initialized Parameters
- G. Servlet Code
- H. Context parameters
 - 1. Adding Context Parameters
 - 2. Servlet Code
- I. Define Session Time Out
- J. Welcome pages
- K. Loading servlet on startup
- L. Defining error pages

IX. Architecting Web Applications

- A. Model 1 architecture
- B. Model 2 architecture
 - 1. Model
 - 2. View
 - 3. Controller
 - 4. Advantages of model 2 architecture

X. Security

- A. Security overview
- B. Levels of Java security
- C. Evolution of Java security
- D. Policy files
- E. Permission class
- F. Message digest
- G. Authentication
- H. Certificate
- I. Controlling applications

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Advanced Java Using Galileo Eclipse

Course Outline (cont'd)

XI. Threads

- A. What is a thread?
- B. Thread class
- C. Two ways to obtain a thread
- D. Thread states
- E. Kinds of thread programming
 - 1. Unrelated threads
 - 2. Related but synchronized threads
- F. Mutually exclusive threads
- G. Synchronizing threads
- H. Common locking problems
- I. wait() and notify

XII. Streams

- A. I/O Characteristics
- B. What is a Stream?
- C. Predefined Streams
- D. Java I/O Classes
- E. Writer Abstract Class
- F. Object Serialization
- G. Writing an Object to a File
- H. Reading an Object from a File

XIII. Networking

- A. Networking basics
 - 1. TCP
 - 2. UDP
 - 3. URL
- B. Networking classes in JDK
 - 1. URL class
 - 2. URL connection
- C. How TCP/IP connection works
- D. Making TCP connection
 - 1. Implementing the server
 - 2. Implementing the client

XIV. Distributed Objects

- A. Object serialization
- B. Writing an object to a file
- C. Reading an object from a file
- D. Remote method invocation
- E. Client/Server communication
- F. Implementing RMI
 - 1. On the RMI server
 - 2. On the RMI client

- 3. Deployment steps

XV. Overview of J2EE Architecture

- A. J2EE architecture
- B. J2EE modules
- C. HTTP servlet
- D. JavaServer pages
- E. Model view controller
- F. Struts
- G. JSTL
- H. EJB
- I. J2EE services
 - 1. Naming service
 - 2. Database access service
 - 3. Transaction service
 - 4. Messaging service
 - 5. JavaMail service
 - 6. Security service

XVI. Overview of EJB

- A. J2EE containers
- B. What is EJB?
- C. EJB framework components
 - 1. EJB server
 - 2. EJB container
 - 3. EJB components
 - 4. EJB deployment descriptors
- D. Roles
- E. Benefits of EJB
- F. When to use EJB
- G. Types of EJB
 - 1. EJB session Beans
 - 2. EJB entity beans
 - 3. Message driven beans