

Java/Servlet/JSP Programming using RAD7 Workshop **Fast Track Class** CDT720

This FAST TRACK 5-day course includes selected topics from Caliber's *Java Programming*, *Servlet Programming*, and *JSP Programming* courses. Some prior knowledge of Java Programming is assumed (see prerequisites below.) All coursework will be done with RAD 7, but no prior knowledge of RAD 7 is assumed. Emphasis will be on object oriented programming, arrays, exceptions. JUnit, servlet request and response objects, receiving and sending web pages, persisting objects through serialization, cookies, session management and JDBC, and writing very simple JSPs.

Audience

- Developers who will be developing browser-based java applications.

Prerequisites

- Some prior knowledge of Java syntax including declaring and initializing primitives, performing simple arithmetic, coding selection and iteration constructs (*if*, *for*, *while*, and *do*), and printing with *System.out.println*.

Course Length

- Five days
-

Learning Objectives

- Write short Java applications (2 days)
- Write web-based servlets (2½ days)
- Short introduction to JSPs (½ day)

Teaching Methods

- Lecture
 - Examples
 - Hands-on exercise
-

BEGIN JAVA PROGRAMMING

Object Oriented Concepts

- Objects vs. classes
- Attributes (fields)
- Behaviors (methods)
- Implementation vs. Interface
- Private vs. public vs. protected
- Constructors: default & overloaded
- Accessor methods
- Mutator methods
- final

Advanced OO Concepts

- Comparing objects
- Cloning objects
- Composition
- Inheritance
- Polymorphism
- Abstract classes and methods

Arrays

- Arrays of primitive data types
- Arrays of objects
- The Vector class

Exceptions

- try and catch blocks
- finally
- Effect of System.exit() within catch
- Creating your own exception
- Throwing the exception
- Handling the exception
- RuntimeException
- Nested exceptions
- The printStackTrace() method

Files

- Writing objects to a file using serialization
- Reading objects from a file written with serialization

JUnit

- What is JUnit
- Unit and regression testing framework

BEGIN SERVLET PROGRAMMING

WAS Server

- Creating a test server

Life Cycle of a Servlet

- Creating a Web Application
- init, doGet, doPost, and destroy() methods

Parameters

- Creating servlet parameters
- Accessing servlet parameters
- Creating context parameters
- Accessing context parameters

Creating a web page using RAD 7

- HTML/XHTML
- Design, Source, and Preview views
- Easy web page creation with Design view
- Form
- Label
- Text field
- List box
- Radio button
- Checkbox
- Submit button
- method = "Get" vs. method = "Post"

Accessing form parameters

- getParameter()
- Detecting null input
- Data validation techniques
- Exception handling

Java Beans

- Defining entity beans (note: not EJBs)
- Instantiating entity beans with form data

Object Persistence

- Data Access Object (DAO) design pattern
- Using serialization
- Using cookies
- Using session managed persistence
- Using JDBC

BEGIN JSP PROGRAMMING

Introduction to JSP Programming

- What is a JSP?
- JSPs: When and why?
- Components of a JSP

Our First JSP

- Creating a JSP
- Including an expression
- Running a JSP on the server

Scripting

- Scriptlets
- Comments, expressions, and declarations

Life Cycle of a JSP

- jspInit()
- jspDestroy()
- <body></body>

Simple JSPs

- Text fields and request.getParameter()
- Submit button
- method = "Get" vs. method = "Post"