



## Java 5 Programming Essentials for Server-Side (J2EE) OO Developers (TT5140)

**Duration:** 5 days

**Course Description:** This comprehensive hands-on training course teaches essential Java 5 programming skills in preparation for building server-side applications on the J2EE platform. This course is geared for software developers who need to understand what J2EE is, what it means in terms of today's systems and architectures, and how to apply J2EE technologies and practices in implementing and deploying robust, flexible web applications. In addition to the normal labs is a case study that covers the entire spectrum from use cases to object-oriented design to implemented classes. This case study supplements the course and can be used during and after the course as a reference and a tool for reviewing and practicing what was learned in class.

**Audience:** Experienced C#/C++ or other OO developers

**Prerequisites:** This is an **introductory level** training course, designed for **experienced OO developers** (such as C++ or SmallTalk) that need to extend their skills in web development and understand the basic architecture of a J2EE web application and/or implement a web application taking advantage of what J2EE brings to the table.

### Topics:

#### PART ONE: ESSENTIALS JAVA PROGRAMMING

##### SESSION: JAVA, A FIRST LOOK

- Using the JDK
- Setting Up Environment
- The Development Process
- Locating Class Files
- Compiling Package Classes
- Source and Class Files
- Applications and Applets
- Writing a Simple Class
- Classes in Java™
- What Is a Class?
- Defining the Class
- Class Modifiers
- Class Instance Fields
- Instance Fields Diagram
- Primitives vs. Object References
- Creating Objects
- The main Method
- Using the Dot Operator
- Writing Output
- The Java™ Platform
- Defining Java

- Java Provides Several Platforms
- Note on Terminology
- Java SE 5
- Java SE Development Kit (JDK)
- Executing Programs
- Lifecycle of a Java Program
- Responsibilities of JVM
- Java is Dynamic: The Runtime Process
- Primary Areas of the JVM Runtime
- Garbage Collection
- Documentation and Code Reuse
- JavaDoc Provides Documentation Delivery
- In-Line Comments are Translated into HTML Rendering
- Working with Java in Your Environment

##### SESSION: OBJECT-ORIENTED CONCEPTS IN JAVA

- Object-Oriented Programming
- The Object Oriented Way
- Real-World Objects
- Classes and Objects
- Examples of Classes and Objects
- Classes and Objects Diagram

## Java 5 Programming Essentials for Server-Side (J2EE) OO Developers (TT5140)

- Object Behavior
- Methods and Messages
- Inheritance, Abstraction, and Polymorphism
- Encapsulation
- Inheritance
- Method Overriding
- Aggregation
- Type Abstraction – Grouping as Supertype
- Polymorphism
- Polymorphism Diagram
- Formatter – Probable First Encounters
- Formatter – StringBuffer
- Specializing in a Subclass
- Extending a Class
- The extends Keyword
- Casting
- Overriding Superclass Methods
- Method Overriding Diagram
- Calling Superclass Methods from Subclass
- The Object Class
- The equals Method
- Default Constructor
- Implicit Constructor Chaining
- Passing Data Up Constructor Chain
- A Common Programming Mistake
- Editing Tools in Your IDE

### SESSION: GETTING STARTED WITH JAVA™

- Adding Methods to the Class
- Instance Methods
- Passing Parameters Into Methods
- Returning a Value From a Method
- Overloaded Methods
- Overloaded Methods Diagram
- Constructors
- Defining a Constructor
- Optimizing Constructor Usage
- Language Statements
- Operators
- Comparison and Logical Operators
- Looping: The for Statement
- Looping: The while Statement
- Looping: The do Statement
- Continue and Break Statements
- The switch Statement
- Using Strings
- Strings
- String Method
- String Equality
- StringBuffer
- Strings, StringBuffer, and StringBuilder
- StringTokenizer
- Scanner
- Scanner - File Source
- Scanner - Getting Input
- Scanner - Testing for Tokens
- Scanner - Patterns for Tokens
- Formatter

### SESSION: ESSENTIAL JAVA™ PROGRAMMING

- Fields and Variables
- Fields vs. Variables
- Data Types
- Default Values
- Block Scoping Rules
- Using this
- Final and Static Fields
- Static Variable Diagram
- Using Arrays
- Arrays
- Accessing the Array
- Multidimensional Arrays
- Static Methods and Fields
- Static Fields
- Simple Example of Static Fields
- Static Methods
- Java™ Packages
- The Problem
- Packages
- Class Location of Packages
- The Package Keyword
- Importing Classes
- Executing Programs
- Visibility
- Java Naming Conventions

## Java 5 Programming Essentials for Server-Side (J2EE) OO Developers (TT5140)

- Packages Diagram
- Refactoring in Your IDE

### SESSION: ADVANCED JAVA™ PROGRAMMING

- Inheritance and Polymorphism
- Polymorphism
- Polymorphism: The Subclasses
- Treating Derived Classes as the Superclass
- Casting to the Derived Class
- Using instanceof For Downcasting
- Upcasting vs Downcasting
- Calling Superclass Methods From Subclass
- The final Keyword
- Interfaces and Abstract Classes
- Separating Capability from Implementation
- Abstract Classes
- Shape as an Abstract Class
- Polymorphism With Abstract Classes
- Interfaces
- Implementing an Interface
- Extending Interfaces
- Polymorphism With Interfaces
- Type Checking
- Abstract Classes vs. Interfaces
- Interfaces Diagram
- Exceptions
- What is an Exception?
- Exception Architecture
- Handling Exceptions
- The Throwable Class
- The try Block
- The catch Block
- The finally Block
- Full Example of Exception Handling
- Generalized vs. Specialized Exceptions
- Overriding Methods
- Creating Your Own Exceptions
- Throwing Exceptions
- Re-throwing an Exception
- Checked vs. Unchecked Exceptions
- Debugging in Your IDE

### SESSION: JAVA™ DEVELOPER'S TOOLBOX

- Utility Classes
- Wrapper Classes
- The Number Class
- Numbers and Strings
- Big Decimal
- Random Numbers
- Decimal Formatting
- The Date Class
- Vector and Hashtable
- The Vector Class
- Creating and Using a Vector
- Java Collections Methods in Vector
- Hashtables
- Understanding How Hashing Works
- Creating and Using a Hashtable
- Performing Lookups
- Collections
- The Collections Framework
- Collections Feature Types
- Collections Interface Hierarchy
- Map Interfaces
- Optional Methods
- The Collection Interface
- Iterators
- The Set Interface
- SortedSet
- Set and SortedSet Example
- Comparable and Comparator
- The List Interface
- List Example
- ListIterator
- Queue Interface
- QueueExample
- BlockingQueue
- BlockingQueue Implementations
- Collections Utility Methods
- Features of the Implementation Classes
- Synchronization Wrappers
- Feature Comparison
- Using the Right Collection
- Use of Collections vs. Vector/Hashtable
- Optimizing Collection Constructors
- Copying Arrays

## Java 5 Programming Essentials for Server-Side (J2EE) OO Developers (TT5140)

- Creating and Using an ArrayList
- Creating and Using a HashMap
- Generics
- Generics and Parametric Polymorphism
- Simple Generics
- The Mechanics of Generics
- Generics and Subtyping
- Compiler Restrictions on Generics and Subtyping
- Generics as Arguments in Methods
- Rationale Behind Wildcards
- Wildcards In Use
- Regular Wildcards in Method Parameters
- Bounded Wildcards
- Standard Rules Apply
- Generic Methods
- Interoperability with LegacyCode
- Raw Types
- Legacy Calls To Generics
- When Generics Should Be Used
- Build Paths in Your IDE

### SESSION: JAVA™ APPLICATION DEVELOPMENT

- What is JDBC?
- Structured Query Language (SQL)
- Connecting to the Database
- Statements
- Statement and PreparedStatement
- ResultSet
- JDBC Diagram
- Executing Inserts, Updates, and Deletes
- Controlling Transactions and Concurrency
- Mapping SQL Types to Java Types
- Database Connection Via JDBC Calls
- Rationale for Connection Pooling
- Connection Pooling in JDBC
- Database Connection Using a DataSource
- Stored Procedures Defined
- Callable Statement Syntax
- Stored Procedure Parameters
- RowSet Implementations

- JDBCRowSet
- JDBCRowSet Approach
- JDBCRowSet – Retrieving Data
- JDBCRowSet Example
- CachedRowSet
- CachedRowSet Approach
- CachedRowSet Example

### PART TWO: ESSENTIAL WEB DEVELOPMENT WITH SERVLETS/JSPS

#### SESSION: J2EE APPLICATION ARCHITECTURE

- Technical Overview of J2EE
- What is J2EE?
- Common Themes In J2EE Framework
- J2EE Containers and Components
- Servlets
- Java Server Pages (JSP)
- EJBs and Web Services
- J2EE Containers
- J2EE Application Modules
- The J2EE 1.4 Specification
- J2EE Platform Roles
- J2EE Application Architectures (web based)
- J2EE and the Web
- J2EE 1st Generation DB-centric Web Application
- Adding JSPs to Separate Presentation
- Eliminating Java code from view
- The Model 2 Architecture
- Using EJBs
- J2EE Infrastructure Supporting Web Applications

#### SESSION: WEB APPLICATIONS

- Understanding Web Applications
- J2EE Application Modules
- The Truth about Archives
- Enterprise Application Archive (EAR)
- Enterprise JavaBean Archive (JAR)
- J2EE Application Client (JAR)
- Resource Adapter Archive (RAR)
- Web Application Archive (WAR)
- Directory Structure

## Java 5 Programming Essentials for Server-Side (J2EE) OO Developers (TT5140)

- Configuring Web Applications
- Mapping an HTTP Request to a Resource
- The web.xml File
- Structure
- Declaring Servlets and JSPs
- Servlet Mapping
- Servlet Init Parameters
- Web Application init Parameters
- Welcome Page
- Error Page
- HTTP Request Information
- Several Options for Sharing Data
- Configuration and Context
- Servlet Variables
- HttpServlet Request
- Threading and Data
- Threading and Data 2
- Client Side Control
- Output Buffering
- Setting Status Codes; Setting Headers
- sendRedirect
- Disabling Client Caching
- Supporting Persistent Connections
- Setting Content Length
- Dynamic Content Pushing
- Maintaining Client State – Sessions
- Session Management
- Tracking Problem – Stateless HTTP
- Data Problem – Session Data
- Solving the Tracking Problem
- Cookies
- Cookie Behavior; Retrieving Cookies
- servlet to set Cookies
- Servlet to Show Cookies
- URL Rewriting
- Solving the Data Problem
- Web Container Manages Session Instances
- Sessions with Cookies
- Cookie-Based Sessions
- Basic Session Implementation
- Cookie Detection is not Standardized
- Getting Rid of Http Sessions
- Session with URL Rewriting

### SESSION: DEVELOPING SERVLETS

- Introduction to Servlets
- Servlet Overview
- Life Cycle of Servlets
- Servlet Lifecycle is Handled by Web Container
- HttpServlet
- Writing the init Method
- HttpServlet doXXX Methods
- Writing a Simple HTTP Servlet
- HttpServletRequest Methods
- ServletResponse
- HttpServletResponse
- Servlet I/O Classes
- Return a Status Code
- Building the Output Document
- Sending Binary Content
- Processing Input Data
- Form Processing with Servlets
- HTML Form
- LoginServlet doPost
- Sever-Side Control
- Request Dispatcher
- Forward the processing
- Passing Processing on and Getting it Back
- Servlet Runs Within Web Container Environment
- Several Options for Receiving Data
- Init Parameters and Attributes
- ServletConfig; ServletContext
- Servlet Variables are Scoped

### SESSION: FILTERS

- Overview of Filters
- What is a Filter
- Single Filter
- Filter Objects
- doFilter Method
- init Method
- Filter Life Cycle
- Cascading Filters

## Java 5 Programming Essentials for Server-Side (J2EE) OO Developers (TT5140)

- Filtering Requests and Responses
- Request Wrapper
- Process the Request
- Examples of Request Filters
- Filter the Response
- Response Wrapper

### SESSION: DEVELOPING JAVASERVER PAGES

- Introduction to JavaServer Pages
- Separating Presentation from Model
- Java Server Page (JSP): An Extension of Servlet
- Lifecycle of a JSP
- Example JSP
- JSP Syntax Consists of Three Types
- JSP Scripting: Declarations; Expressions; Scriptlets & Directives
- The session Attribute
- The errorPage/isErrorPage Attribute
- JSP Actions
- JSP Actions: Include/Forward
- Typical JSP Access Model
- JSP Action: useBean
- Implicit Objects
- JSPs or Servlets?
- JSP Implicit Objects
- Implicit Objects
- Page Object; Config Object
- Request Object; Response Object
- Out Object
- Output Buffer
- Session Object
- Application Object
- PageContext Object
- Attributes
- Session Attributes
- Exception Handling