



## Java 6 Programming Essentials for Server-Side (JEE) Developers (NETD5140-J6)

**Length:** 5 days

### Description

Java Enterprise Edition (JEE) is a powerful platform for building web applications. The JEE platform offers all the advantages of developing in Java plus a comprehensive suite of server-side technologies. This course conveys what you need to know to design and build your own web applications. You'll learn the details of the key JEE technologies and how to leverage the strengths of each, with special focus on Servlets and JSPs. At the same time, you'll be learning about the big picture of JEE and how to design web applications that are robust, efficient, secure, and maintainable.

### Audience

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 JEE web application and/or implement a web application, taking advantage of what JEE brings to the table.

### Prerequisite

Experienced OO developers

### Topics

#### PART ONE: ESSENTIALS JAVA PROGRAMMING

##### 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 6
- 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

## Java 6 Programming Essentials for Server-Side (JEE) Developers (NETD5140-J6)

- JavaDoc Provides Documentation Delivery
- In-Line Comments are Translated into HTML Rendering
- Working with Java in Your Environment
- Looping: The for Statement
- Looping: The while Statement
- Looping: The do Statement
- Continue and Break Statements
- The switch Statement
- Using Strings

### 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
- Object Behavior
- Methods and Messages
- Inheritance, Abstraction, and Polymorphism
- Encapsulation
- Inheritance
- Method Overriding
- Aggregation
- Type Abstraction – Grouping as Supertype
- Polymorphism
- Polymorphism Diagram
- 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
- 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 Subcla
- 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

### 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

### ESSENTIAL JAVA PROGRAMMING

- Fields and Variables



## Java 6 Programming Essentials for Server-Side (JEE) Developers (NETD5140-J6)

- 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
- Packages Diagram
- Refactoring in Your IDE
- 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

### 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

### 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



## Java 6 Programming Essentials for Server-Side (JEE) Developers (NETD5140-J6)

- 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
- 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 Legacy Code
- Raw Types
- Legacy Calls To Generics
- When Generics Should Be Used
- Build Paths in Your IDE
- Overview of Java GUIs
- JFC – Java Foundation Classes
- Categories of Classes in JFC
- Creating the Frame
- Adding Content to a Frame
- A Closer Look at Layout Managers
- BorderLayout
- JFC Provides an Event Handling Structure
- Autoboxing, Enhanced for Loop and Varargs
- Autoboxing/Unboxing
- Autoboxing/Unboxing Issues
- For() Loops and Collections
- The Enhanced for() Loop
- Another Example - Problematic
- Another Example – Enhanced for()
- Enhanced for() Loop Restrictions
- Previous Variable Argument Support
- Variable Arguments

## Java 6 Programming Essentials for Server-Side (JEE) Developers (NETD5140-J6)

- Varargs rules
  - Issues
  - Enumerations and Static Imports
  - Rationale for Enumerations
  - Enumeration Syntax
  - Enumerations Methods
  - Enumerations as a Better Class Type
  - Enumeration Code
  - EnumSet
  - EnumMap
  - When You Should Use Enumerations
  - Using Static Imports
  - When You Should Use Static Imports
  - Java I/O
  - The Java I/O Mechanism
  - Byte Level I/O
  - Subclasses Accessing Real Data
  - Filter Classes
  - Reading Stream as Data Primitives
  - Adding Buffering for Better Performance
  - Writing Buffered Data to a File
  - Character Level I/O
  - Subclasses Accessing Real Data
  - Reader/Writer Filter Classes
  - Serialization
  - Switching Java Versions in Your IDE
  - Introduction to Annotations
  - Annotations Overview
  - Working with Java Annotations
  - Example of Using @Override
  - Annotations are Heavily Used in Many Technologies
  - Declaring Persistence in Hibernate
  - 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
  - Java Logging
  - Why Logging?
  - Logging Framework
  - Logging in Java
  - Java Logging Framework
  - The Logger Class
  - Global Configuration
  - Logging Levels
  - Programmatically Setting Logging Properties
  - Programmatic Handlers
  - Formatters
  - Logging Security & Performance
- JAVA APPLICATION DEVELOPMENT - JDBC**
- What is JDBC?
  - Structured Query Language (SQL)
  - Connecting to the Database
  - Statements

## Java 6 Programming Essentials for Server-Side (JEE) Developers (NETD5140-J6)

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

#### JEE APPLICATION ARCHITECTURE

- Technical Overview of JEE
- What is JEE?
- Common Themes In JEE Framework
- JEE Containers and Components
- Servlets
- Java Server Pages (JSP)
- EJBs and Web Services
- JEE Containers
- JEE Application Modules
- The JEE 1.5 Specification
- JEE Platform Roles
- Annotations
- Dependency Injection
- JEE Application Architectures (web based)
- JEE and the Web
- Adding JSPs to Separate Presentation
- Eliminating Java code from view
- The Model 2 Architecture
- Using EJBs
- JEE Infrastructure Supporting Web Applications Lesson Review

#### WEB APPLICATIONS

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

#### 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
- Server-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

## Java 6 Programming Essentials for Server-Side (JEE) Developers (NETD5140-J6)

- Servlet Variables are Scoped
  - 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 Management
  - Tracking Problem – Stateless HTTP
  - Data Problem – 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 Instances
  - Sessions with Cookies
  - Cookie-Based Sessions
  - Basic Implementation
  - Cookie Detection is not Standardized
  - Getting Rid of Http Sessions with URL Rewriting
  - Application and Events
  - Event Listener Model
  - Life Cycle Events in a Web-Application
  - Declare the Listener
  - Type of Events
  - Context Listeners; Listeners
  - Listeners for Session-Objects
- FILTERS**
- Overview of Filters
  - What is a Filter
  - Single Filter
  - Filter Objects
  - doFilter Method
  - init Method
  - Filter Life Cycle
  - Cascading Filters
  - Filtering Requests and Responses
  - Request Wrapper
  - Process the Request
  - Examples of Request Filters
  - Filter the Response
  - Response Wrapper
- 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 Attribute
  - The errorPage/isErrorPage Attribute
  - JSP Actions
  - JSP Actions: Include/Forward
  - Typical JSP Access Model
  - JSP Action: useBean
  - Implicit Objects
  - JSPs or Servlets?
  - The web.xml in JEE 5
  - <jsp-config>



## **Java 6 Programming Essentials for Server-Side (JEE) Developers (NETD5140-J6)**

- JSP Format Rules
- JSP Error Pages
- JSP Implicit Objects
- Implicit Objects
- Page Object; Config Object
- Request Object; Response Object
- Out Object
- Output Buffer
- Object
- Application Object
- PageContext Object
- Attributes
- Attributes
- Exception Handling