

TT2101-J6: Core Java 6 Programming Fundamentals - 3 days

Duration: 3 days

Skill Level: Introductory

Focus: Java 6 Applications

Audience: Programmers with prior Object-Oriented exposure and background (C++, etc)

Hands-On: Extensive Hands-On Programming Labs; Expert lecture combined with open discussions and high-level demonstrations

Language / Tools: Java 6 delivered with most IDEs: IBM Rational Application Developer (RAD) 7.5, Oracle JDeveloper, Eclipse Galileo/Helios, MyEclipse and more. Course is also offered for Java 5.

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

Customizable: Yes



Core Java 6 Programming Fundamentals is a three-day, fast-paced, quick start to Java 6 training course geared for developers who have prior working knowledge of object-oriented programming languages such as C++. Throughout the course students learn the best practices for writing great object-oriented programs in Java 6 using sound development techniques, new improved features for better performance, and new capabilities for addressing rapid application development. In addition to the normal exercises that are liberally sprinkled throughout the course, there 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.

► **Course Objectives: What You'll Learn**

Students who attend *Core Java 6 Programming Fundamentals* will leave this course armed with the required skills to develop solid object-oriented applications written in Java, using sound coding techniques and best coding practices.

At the conclusion of the course, attendees will be able to:

- Understand not only the fundamentals of the Java language, but also its importance, uses, strengths and weaknesses
- Understand the basics of the Java language and how it relates to OO programming and the Object Model
- Work with the Java Virtual Machine and understand what functions it performs in running Java applications
- Learn to use Java multi-threading and exception handling features
- Understand and use classes, inheritance and polymorphism
- Work with various simple and complex data constructs as well as fields and methods
- Understand and use collections and generics including new Java 6 features and capabilities
- Take advantage of the Java tooling that is available with the programming environment being used in the class

Throughout the three-day 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.

► **Audience & Pre-requisites: Who Should Attend**

This is a **beginner level** Java programming training course, designed for experienced developers who wish to get up and running with Java, or who need to reinforce sound Java coding practices. Attendees should have a working knowledge of developing OO software applications.

► **Related Courses: Suggested Learning Path**

Take Before: Students should have skills equivalent to or should have taken **TT1130 Object-Oriented Analysis & Design Using UML 2.x** as a pre-requisite.

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

- OO developers who want a more in-depth exploration of Java should consider our 5 day superset of this course, **TT2100-J6 Java 6 Programming Fundamentals**
- Developers with little or no prior OO development background, who wish to learn basic Java, may wish to take our 5 day **TT2120-J6 Java 6 Programming Fundamentals for Non OO Developers**
- Programmers preparing for non-GUI, server side / J2EE web development work might want to consider **TT5140-J6 Java 6 Programming Essentials for Server-Side OO Developers** which combines basic Java and essential J2EE (Servlets/JSPs) content.
- Please note that all Java essentials courses are also available using Java 5.

Take After: We offer a variety of introductory through advanced

development, project management, engineering, architecture and design courses.

- Students needing an essential J2EE follow up may take **TT5100 Mastering J2EE (or JEE) Web Applications** (Servlets/JSPs, Tags, JDBC, Security, etc.)
- Essential or Advanced Java topics or basic J2EE / JEE training
- Object-oriented or design patterns topics
- Java Security topics
- Agile or test-driven development topic
- Service-Oriented Analysis and Design
- Web Services – Intro through Advanced
- AJAX, XML or other Web Development topics

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 Java development skills, coupling the most current, effective techniques with the soundest coding practices.

As a basic programming class, this course provides **over 15 basic and challenges labs** for students to work through during the class. This workshop is about **50% hands-on lab 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 Java 6 compliant software combinations, including but not limited to: Eclipse Helios/Galileo, MyEclipse, IBM® WebSphere Rational Application Developer (RAD7.5), Oracle JDeveloper or other IDEs. This course is also offered in Java 5. 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.

Workshop Topics Covered

Session Java: A First Look

Lesson Using the JDK

- Setting Up Environment
- The Development Process
- Locating Class Files
- Compiling Package Classes
- Source and Class Files
- Applications and Applets

Lesson 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

Lesson 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
- JavaDoc Provides Documentation Delivery
- In-Line Comments are Translated into HTML Rendering
- Working with Java in Your Environment

Session Object-Oriented Concepts In Java

Lesson 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

Lesson Inheritance, Abstraction, and Polymorphism

- Encapsulation
- Inheritance
- Method Overriding
- Aggregation
- Type Abstraction – Grouping as Supertype
- Polymorphism
- Polymorphism Diagram

Session Getting Started with Java™

Lesson 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

Lesson 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

Lesson 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
- Formatter – Probable First Encounters
- Formatter – StringBuffer

Lesson 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 Essential Java™ Programming

Lesson Fields and Variables

- Fields vs. Variables
- Data Types
- Default Values
- Block Scoping Rules
- Using this
- Final and Static Fields
- Static Variable Diagram

Lesson Using Arrays

- Arrays

- Accessing the Array
- Multidimensional Arrays

Lesson Static Methods and Fields

- Static Fields
- Simple Example of Static Fields
- Static Methods

Lesson 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

Session Advanced Java™ Programming

Lesson 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

Lesson 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

Lesson 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

Lesson Utility Classes

- Wrapper Classes
- The Number Class
- Numbers and Strings
- Big Decimal
- Random Numbers
- Decimal Formatting
- The Date Class

Lesson 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

Lesson 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

Lesson 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

Lesson 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
- Switching Java Versions in Your IDE

LABS: There are over 20 hands-on lab exercises and challenges laced throughout this course. Please ask us for lab details.

Case Study: A complete case study provides an incremental series of labs covering requirements analysis, design, implementation, and refactoring. A complete set of solutions are also included, enabling the case study to be used after class as a homework assignment, tool for review, or a reference.

Need more details? 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.

► Easy 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 solutions to ensure your needs are met, whether we customize the material, or devise a different educational path to prepare for this course.

► 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.

► Why Work with Trivera Technologies?

- **We provide a solid Java and OO development foundation.** Students will learn how to develop (and reuse!) essential Java 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 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 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 J2EE 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.
- **True content ownership gives us flexibility & quality above the rest.** These course materials are wholly-owned by our company and fully customizable - at little or no cost- to help you best meet your learning objectives. We have many dedicated experts available worldwide to instruct your team, and can provide services around the globe, either locally or virtually. We work closely with you to produce the most effective events and materials for your team, within your desired timeframe and budget.
- **We have to adhere to higher standards.** As a courseware provider, 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, J2EE, JEE / JavaEE, .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.
- **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 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 produced virtually for a **private presentation**, 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 for corporate license with complete instructor support and free corporate branding. 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 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.