

## **Introduction to Java Programming for Mainframe Developers**

- **Course Number:** IntJava-MF
- **Length:** 5 Day(s)

### **Certification Exam**

There are no exams associated with this course.

### **Course Overview**

This course uses a combination of instructor lectures, computer demonstrations and interactive hands-on lab simulations to illustrate the major parts of the course. Topics discussed include: Mainframe vs eCommerce Model and the evolution of the eCommerce Model. We will then discuss J2EE basics, languages & tools, OO concepts, classes, types and objects. We will discuss core concepts, methods, constructors, class data & security and inheritance hierarchies. We will then move on to learning more about the java environment and discussing the Eclipse IDE. We will spend time defining & declaring variables, talking about decision logic & looping, illustrating class definitions and java arrays. We will define inheritance, talk about the eclipse debugger, exception handling, database concepts, thread management, and wrap things up with a discussion on files and streams.

### **Prerequisites**

Basic knowledge of programming and Java programming language is required.

### **Audience**

This course is intended for software developers using Java programming language.

### **Course Outline**

- Level 1
- Mainframe vs eCommerce Model
- Mainframe Model
- Mainframe Architecture
- Components & Containers
- Model-View-Controller
- eCommerce Model
- MVC Approach
- Mainframe to MVC Comparison
- Java Components/Containers
- Mainframe vs. eCommerce
- Mainframe Containers
- eCommerce Server Model
- Message Components
- Terminal Emulation
- Message Components
- HTML vs MFS/BMS
- HTML FORM Source
- Message Components
- Web Server

- Message Components
- Web & Application Server
- Message Components
- eCommerce Model evolution
- eCommerce circa 1995
- eCommerce evolution
- Development Paths
- Evolution of Servlets
- J2EE Basics
- Java Platform Separation
- J2EE Extension API
- J2EE Platform
- J2EE Architecture
- Level 2
- Languages & Tools
- Defining Object Orientation
- Object-Oriented vs Procedural
- Procedural Programming
- Object Oriented Programming
- Where's the Code
- OO Technologies
- Comparing OO Languages
- ToolKits and Frameworks
- OO Distributed Development
- OO Components
- Enterprise Concepts
- OO Methodologies
- Commercial OO Methodologies
- OO Modeling
- OO Evolution
- Procedural Language Problems
- Object Orientation over Procedural
- Object Oriented Advantages
- Object Oriented Disadvantages
- OO Concepts
- OO Fundamentals
- Defining Objects
- Object Illustration
- Object Characteristics
- OO Object Samples
- Object Building Blocks
- Concept of Templates
- What are Instances
- OO Application Composition
- Object Lifecycle
- Classes, Types and Objects
- OO Methods and Variables
- Defining OO Methods
- Defining OO Variables
- OO Methods
- Messaging Example
- Object Architecture

- Java Class Example
- Use of private
- Class Definition
- Variable Scope
- Identifiers vs Values
- OO Static Typing
- Core Concepts
- Major OO Aspects
- Defining Encapsulation
- Encapsulation Summary
- Defining Inheritance
- Illustrating Inheritance
- What is a Superclass
- Defining Subclass
- SubClass Inheritance
- Polymorphism
- Illustrating Polymorphism
- Review
- Level 3
- Methods
- Method Overloading
- Method Overriding
- Constructors
- Building New Objects
- Defining Constructors
- Sample Constructor
- Constructor Chaining
- Class Data & Security
- Class Versions
- Access Control Modifiers
- Class Access Options
- Inheritance Hierarchies
- What is Class Inheritance Derived Class Objects Illustrating Inheritance Inheritance Specification Using Derivation Utilize Inheritance Defining Polymorphism What are Interfaces
- Level 4
- Learning the Java Environment
- What is the Java Language?
- Java Language Background
- Java Language Benefits
- Types of Java Programs
- Defining a Java Applet
- Defining a Java Application
- Defining a Java Servlet
- Java Runtime Environment
- Java Development Environment
- Biggest Benefit: Portability
- Portability Benefits
- The Java Language
- Java Libraries
- Java API Hierarchy
- Object-Oriented vs. Procedural
- Language Attributes

- History of OO Programming
- Java and OO Concepts
- OO Concept: Abstraction
- OO Concept: Encapsulation
- OO Concept: Polymorphism
- OO Concept: Inheritance
- What are Objects?
- OO Programming in Java
- Object Architecture
- Objects and Classes
- Viewing a Class Definition
- Creating Java Class Definition
- Data Hiding in Java
- Inheritance: Class & Subclasses
- OO Program Structure
- New Objects: Constructors
- Java Platform Separation
- J2EE Architecture
- Illustrating J2EE
- J2SE Platform
- J2EE Packaging
- Demo - Eclipse SDK 3.1
- Chapter 4 Review
- Level 5
- Eclipse IDE
- Understanding Eclipse
- Eclipse Architecture
- Java Development
- Eclipse Window
- Workbench Window
- Workbench Wizards
- Role of Eclipse Projects
- Project Creation
- Eclipse Folder Structure
- Java Project Folder
- Demo - Exploring Eclipse
- Importing Projects
- Demo - How to Import Projects
- Eclipse Perspective
- Utilizing Perspectives
- Selecting Perspectives
- Java Perspective
- Java Browsing Perspective
- Debug Perspective
- Demo - Perspectives
- Common Views: Navigator & Editor
- Eclipse Prefs & Java Code Formatting
- Workbench Preferences
- Managing File Editors
- Workbench Preferences: File Editors
- Demo - Eclipse Preference Settings
- Illustrating Java Editor

- Search Operations
- Java IDE
- Eclipse Java Decomposition
- Incremental Compilation
- Illustrating Debugger Role
- Demo - Eclipse Debugger
- Workbench Exports
- Lab - Working with Eclipse
- Chapter 5 Review
- Level 6
- Defining & Declaring Variables
- Java Variables
- Java Primitive Data Types
- Java Integer Data Types
- Integer Internal Representation
- Declaring Integer Variables
- Assignment Statement
- Demo - Integer Variable Types
- Arithmetic Calculations: Binary Operator
- Arithmetic Calculations: Unary Operator
- Integer Division with Remainders
- Increment and Decrement
- Implicit Type Conversion
- Explicit Casting
- Floating Point Data Types
- Floating Point Variables
- Floating Point Calculations
- Using the Modulus Operand
- Explicit Casting: Floating Point
- Storing Characters in Java
- Character Arithmetic
- Boolean Variables
- String Variables
- Lab - Defining Variables
- Chapter 6 Review
- Level 7
- Decision Logic & Looping
- Java Comparison Operators
- Comparison Operators and Booleans
- Simple if Statement
- Use of Statement Blocks
- True/False using else
- Sample else Conditional
- Completed Java Program
- Statement by Statement Examination
- Nested if Statements
- Demo - Java Conditional
- Boolean Operators
- Using Conditional AND
- Conditional OR Operand
- Demo - Conditional AND
- Using the NOT Operand

- Booleans in Combinations
- Using Conditional Operators
- Example of Ternary Operators
- The continue Statement
- Use of switch Expression
- Using switch Option
- Switch Without break Statements
- Stacking case Statements
- Demo - Switch Case Statement
- Using the for Loop in Java
- Example of the for Loop
- Sample of a for Loop
- Demo -for Loop
- Using the while Loop in Java
- Example of the while Loop
- Sample of a while Loop
- Using the dowhile Loop
- Example of the dowhile Loop
- Sample of a dowhile Loop
- Nesting Iterative Loops
- Use of the continue Statement
- Example of the continue Statement
- Using Labels with continue
- Use of the break Statement
- Example of the break Statement
- Demo - Iterative Processing
- Lab - Conditionals
- Chapter 7 Review
- Level 8
- Illustrating Class Definitions
- Illustrating a Class
- Class Variables
- Instance vs Class Variables
- Methods and Class Definitions
- Class Definition
- Defining Methods
- Parameter Lists
- Method Calls in Static Methods
- How Arguments are Passed
- Accessing Methods & Variables
- Class Method Definitions
- Instance Method Definitions
- Initialize Instance Variables
- Use of Initialization Blocks
- Defining a Constructor
- Create Objects with Constructor
- Declaring Variables vs Objects
- Demo - Employee Class Definition
- Demo - Employee Constructor
- Using Constructors
- Passing Objects to Methods
- Demo - Passing Java Objects

- Constructors & Method Overloading
- Multiple Vehicle Constructors
- Using Multiple Constructors
- Constructor to Constructor Calls
- Understanding Java Packages
- Compiling using Packages
- Using Java Extensions
- Creating .jar Files
- Using Package Classes
- Class Access Options
- Class Access within Package
- Access from Different Packages
- Demo - Using Import Statement
- Lab - Class Definition
- Chapter 8 Review
- Level 9
- Java Arrays
- What is an Array?
- Declaring an Array
- Accessing Array Elements
- Using an Array
- Demo - Defining & Initializing an Array
- MultiDimensional Arrays
- Sample MultiDimensional Array
- Lab - Java Arrays
- Chapter 9 Review
- Level 10
- Defining Inheritance
- Defining Derivation
- Using Derivation in Java
- What is Class Inheritance
- Inherited Methods
- Derived Class Objects
- Class Derivation Example
- SubClass Constructors
- Overriding Base Methods
- Base Class Access Attributes
- Demo - Java Inheritance
- What is Polymorphism?
- Polymorphism Conditions
- Using Polymorphism
- Execute Polymorphic Application
- View of Derived Object
- Demo - Polymorphism
- Abstract Classes
- Abstract Example
- Use of final with Methods
- Use of final with Class
- Universal SuperClass Concept
- Object Protected Methods
- Using toString() Method
- getClass Inherited Method

- What are Interfaces
- Defining Class Interfaces
- Benefits of Interfaces
- Implementing Interfaces
- Implement Interface Methods
- Summary of Interfaces
- Demo - Interfaces
- Lab - Inheritance
- Chapter 10 Review
- Level 11
- Eclipse Debugger
- Illustrating the Debugger Role
- Debugging in WSAD
- Debugger Features
- Debug Panel
- Demo - Debugger Overview
- Setting Debugging Preferences
- Debug Filters
- Debug Preferences
- Debug Preferences: Console
- Demo - Eclipse Debugger Settings
- Starting Debugger
- Debug Source Panel
- Suspended Thread
- Controlling Execution
- Viewing Variables
- Using Expressions
- Setting Breakpoints
- Role of Breakpoints
- Viewing Breakpoints
- Breakpoint Properties
- Exceptions Panel
- Lab - Debugger
- Chapter 11 Review
- Level 12
- Exception Handling
- What is an Exception?
- Different Types of Exceptions
- Error Subclasses
- Exception Classes
- RuntimeException Errors
- Exception Handling in Java
- Defining Exceptions to Throw
- Handling Method Exceptions
- Using try Block
- Using catch Block
- Try/Catch Block Example
- Demo - Try/Catch Block
- Try/Catch Block with Loop
- Demo - Try/Catch Block with Loop
- Multiple catch Blocks
- Using finally Block

- Exception Handling Summary
- Demo -finally Block
- Normal Execution Pattern
- Exception Execution Pattern
- Exception Not Caught
- Rethrowing Exceptions
- Throwable Class
- Using Throwable Class
- Demo - Exception Types
- Defining an Exception Class
- Throwing your Exceptions
- Lab - Exception Handling
- Chapter 12 Review
- Level 13
- Database Concepts
- JDBC Accessing
- Illustrating JDBC API
- Java Application Perspective
- Role of DBMS Driver
- Thick vs. Thin JDBC Drivers
- Type 1: Bridge Driver
- Type 2: Native API
- Type 3: Network Driver
- Type 4: Native Driver
- Comparing JDBC Versions
- JDBC Architecture
- Build Connection Object
- Basic JDBC Logic Flow
- Defining a Database Driver
- Demo - Defining a Connection Object
- DriverManager in JDBC
- Connecting to a Datasource
- Sample Connection
- Username/Password
- Defining a Statement
- Demo - Statement Object
- Prepared SQL Statement
- Exploring ResultSet Object
- Accessing ResultSet Object
- Demo - ResultSet Object
- Processing ResultSet Table
- JDBC Data Type Conversions
- Demo - ResultSet Processing
- Lab - JDBC Access
- Chapter 13 Review
- Level 14
- Thread Management
- What is a Thread
- Thread Utilization in Java
- Illustrating Threads
- Java Thread Illustration
- Thread Illustration View

- Thread Lifecycle
- Thread Creation
- Demo - Thread Execution
- Thread Controls
- Thread Control Methods
- Demo - Thread Interruption
- Define Thread Subclass
- Sample of Thread Subclass
- Thread Characteristics
- Daemon and User Threads
- Creating Thread Objects
- Implementing the run method
- Stopping a Thread
- Additional Thread Methods
- Demo - Runnable Interface
- MultiTasking vs MultiThreading
- Sample Non-Threaded Example
- Sample Thread Example
- Thread Synchronization
- Synchronize Example
- Thread Scheduling
- Illustrating Synchronized
- Defining Deadlocks
- Lab - Threads
- Chapter 14 Review
- Level 15
- Files and Streams
- What is a Stream?
- Stream Input/Output Operations
- Defining a File
- Testing File Objects
- Sample Using File
- Demo - File Object Creation
- Accessing File Objects
- Demo - File Handling
- Modifying File Objects
- Using Directory Methods
- Using Attribute Methods
- Using Creation Methods
- Using Deletion Methods
- Demo - File Directory
- Subclasses of OutputStream
- Defining OutputStream Methods
- Using FileOutputStream Class
- Using the FileOutputStream
- ByteArrayOutputStream
- Demo - Data Output Stream
- Using DataOutputStream Class
- Buffered Output Stream
- Create Buffered Output Stream
- Demo - Buffered Ouput
- Lab - File Management

- Chapter 15 Review
- Course Closure