Introduction To Java 8 Using Eclipse

Duration: **5 days**

Overview:

This course introduces the Java programming language and how to develop Java applications using Eclipse. Students learn the syntax of the Java programming language, object-oriented programming using Java, exception handling, generics, collections, and file input/output (I/O). During the course, students will develop and test Java applications using Eclipse.

Target Audience:

This course is intended for programmers who are interested in adding Java to their skills.

Pre-requisites:

Before attending this course, students must be familiar with object-oriented principles and the concept of object-oriented programming.

Module 1: Overview of Java

- History Of Java
- Benefits Of Java
- What Is Java?
- What's This "Virtual Machine"?
- Comparison to Other Languages
- Java Programs
- Basic Java Development Tools
- Java Editions
- Example – HelloWorld
- Java Classes
- Main Methods
- Statements
- Summary

Module 2: Java Tools in Eclipse

- Eclipse Platform
- Eclipse Workspace
- Perspectives, Views & Editors
- Basic Operations with Eclipse Views and Perspectives
- The Java Perspective
- The Debug Perspective
- Navigator View
- Package Explorer
- Outline View
- Problems View
- Eclipse Preferences
- Build and Validation
- Code Completion, Templates and Snippets
- Searching
- Configure Compiler Class Path
- JRE Switching

Module 3: Basic Object Concepts

- What Is An Object?
- State
- Behavior
- Encapsulation
- Encapsulation Examples
- Classes vs. Objects
- Inheritance
- Interfaces
- Polymorphism
- Benefits Of Objects
- Summary

Module 4: Basic Java Syntax

- Declaring And Initializing Variables
- Keywords
- Coding Tips – Variables
- Primitive Data Types
- Logical - boolean
- Textual - char and String
- Integral - byte, short, int, long
- Floating Point - float and double
- Literal Values
- Java 7 – Changes in Numeric Literals
- Strings
- Creating Strings
- White Space
- Comments
- Coding Tips - Comments
- Java Statements
- Coding Tips - Statements
- Scope of a Variable
- System.out/System.in
- Scanner Class
- Summary

Module 5: Operations and Making Decisions

- Operator Categories
- Special Situations
- Binary Operators
- Integer Division
- Numeric Promotion
- Type Conversion Of Primitive Types
- Unary Operators
- Relational Operators
- Logical Operators
- "Short Circuited" Operators
- Bitwise Operators
- Bitwise Examples
- Shift Operators
- Overflow And Underflow
- Assignment Operators
- Ternary Operator
- Calculation Errors
- Operator Precedence
- Precedence Examples
- Combining Strings
- Coding Tips - Operators
- Control Flow Statements
- 'if' Statement
- 'if…else' Statement
- Nested Statements
- Coding Tips - if & if-else
- Summary

Module 6: Using Classes and Objects

- Objects, Instances, And Classes
- What Are Classes?
- Working With Classes And Objects
- Instantiation
- Instance Methods
- Object References
- String Operations
- "Wrapper" Classes
- Autoboxing
- Summary

Module 7: Writing Classes

- Why Define Your Own Classes?
- Encapsulation
- Elements Of A Class
- Defining Classes
- Coding Tips - Class Definitions
- Fields
- Defining Fields
- Coding Tips - Fields
- Methods
- Defining Methods
- Passing Parameters
- Overloading Methods
- Coding Tips - Methods
- Local Variables vs. Instance Variables
- Example - Defining a Class
- Example - Fields
- Example - Defining a Method
- Example - Calling a Method
- Summary

Module 8: Controlling Code Access and Code Organisation

- Controlling Access
- Data Hiding
- Encapsulation
- JavaBeans
- Packages
- Naming Packages
- Declaring Packages In Classes
- Problems Solved With Packages
- Package Access
- Example - Access Modifiers
- Import Statement
- Using Classes From Packages
- Coding Tips - Import Statements
- Correlation To File Structure
- Class Path
- Java Core Packages
- Java API Documentation
- Summary

Module 9: Constructors and Class Members

- Constructors
- Default Constructor
- Multiple Constructors
- Defining Constructors
- Example - Calling Constructors
- "Good" Constructors
- 'this' Keyword
- Using 'this' to Call a Constructor
- Using 'this' to Set a Field
- Class Members
- Examples Of Class Members
- Comparison With Instance Members
- Use Of Class Variables
- Static Class Methods
- Use Of Class Methods
- The Math Class
- Main Method And Command Line Arguments
- Declaring Constants
- Coding Tips - Class Members
- Useful Standard Class Members
- Initialization Blocks
- Static Initialization Blocks
- Summary

Module 10: Advanced Control Structures

- 'switch' Statement
- Example - switch
- Switch "Fall Through"
- Using switch "Fall Through" for Multiple Options
- Java 7 – Strings in switch Statement
- 'for' Loop
- Example - for
- 'while' Loop
- Example - while
- 'do…while' Loop
- Example - do while
- Break Statement
- Example - break
- Labeled Statements
- Example - Labeled break
- Continue Statement
- Example - continue
- Example - Labeled continue
- Coding Tips - Control Structures
- Summary

Module 11: Inheritance

- Inheritance Is…
- Inheritance Examples
- Declaring Inheritance
- Inheritance Hierarchy
- Access Modifiers Revisited
- Inherited Members
- Instances Of A Subclass
- Example Of Inheritance
- Role In Reuse
- The super Keyword
- Example - super Keyword
- Problems with Constructors
- Limiting Subclasses
- Calling Methods in Constructors
- The Object Class
- Summary

Module 12: Arrays

- Arrays
- Declaring Arrays
- Populating Arrays
- Accessing Arrays
- Arrays of Objects
- Array Length
- Coding Tips - Arrays
- Array References
- Multidimensional Arrays
- Arrays Of Arrays
- Copying Arrays
- For-Each loop
- Variable Arguments
- Variable Arguments Example
- Summary

Module 13: Commonly Overriden Methods

- Overriding Methods
- @Override Annotation
- Using Eclipse to Override Methods
- toString()
- toString() in Object
- Overriding toString()
- Comparing Objects
- Using == vs. equals(..)
- Overriding equals(..)
- Complex Comparisons
- equals(..) Example
- hashCode()
- Overriding hashCode()
- hashCode() Example
- Generating equals and hashCode
- Summary

Module 14: Advanced Java Tools

- Refactoring
- Renaming Elements
- Moving a Class to a Different Package
- Extracting Code to a Method
- Other Source Code Refactoring
- Refactoring to Improve Type Hierarchy
- Generalizing a Variable
- Pull-up and Push-down

Module 15: Exceptions

- What is an Exception
- Benefits
- The java.lang.Exception Class
- How to Work With Exceptions
- Example Exception Handling
- The try-catch-finally Statement
- Flow of Program Control
- Exception Hierarchy
- Checked Exceptions
- Unchecked Exceptions
- Coding Tips - Exception Types
- Catching Subclass Exceptions
- Java 7 – Catching Multiple Exceptions
- Specifying Thrown Exceptions
- Rethrowing Exceptions
- Java 7 – Rethrowing Exceptions
- Chaining Exceptions
- Creating your Own Exception
- Java 7 – try-with-resources Statement
- Java 7 – Suppressed Exceptions in try-with-resources
- Summary

Module 16: Interfaces and Polymorphism

- Casting Objects
- The instanceof Operator
- Abstract Classes
- Abstract Class – An Example
- Interface
- Interface – An Example
- Comparable Interface
- Comparable Example
- Coding Tips - Superclass or Abstract Class/Interface?
- Coding Tips – Abstract Class or Interface
- Polymorphism
- Conditions for Polymorphism
- Coding Tips - Leveraging Polymorphism
- Covariant Return Types
- Covariant Return Types – An Example
- Summary

Module 17: Useful Java Classes

- Java Logging API
- Control Flow of Logging
- Logging Levels
- Loggers
- Logging Example
- Logging Handlers
- Logging Formatters & Log Manager
- Logging Configuration File
- Example Logging Configuration File
- Logging Filters
- java.lang.StringBuilder
- java.util.StringTokenizer
- java.util.Arrays & java.util.Collections
- java.util.Random
- java.util.Date
- GregorianCalendar & Calendar
- Formatting
- Formatting Example
- Summary

Module 18: Collections and Generics

- What are Collections?
- Arrays vs. Collections
- Main Collections Interfaces
- java.util.Collection
- Main Collection Methods
- Sets
- java.util.List
- java.util.Queue
- Iteration on a Collection
- Iterator vs. For-Each Loop
- Maps
- java.util.Map
- Other Maps
- Collections Implementations
- Abstract Implementations
- Choosing a Collection Type
- Generics
- Generics and Collections
- Generic Collection Example
- Collections and Primitive Types
- Generic Diamond Operator
- Summary

Module 19: Input and Output

- Overview of Java Input/Output
- The File Class
- File Example
- Java 7 - The java.nio.file.Path Interface
- Serialization
- Serializing Object State
- Avoiding Serialization Problems
- serialVersionUID
- Options for File Input/Output
- Streams
- Input Stream
- Output Stream
- "Chained" Streams
- RandomAccessFile
- Java 7 – try-with-resources Statement
- Using Streams - Write Example
- Using Streams - Read Example
- Reader and Writer
- Using Readers and Writers - Write Example
- Using Readers and Writers - Read Example
- Using Readers and Writers - Scanner Read Example
- NIO Channels and Buffers
- Summary

Module 20: Other Java Concepts

- Annotations
- Enumerated Types
- Enumerated Types – Example
- Assertions
- Assertions Example
- When to use Assertions
- Enabling Assertions
- JVM Storage Areas
- Java Heap Space
- Heap Size Limits
- Garbage Collection Basics
- Allocation Failure (AF)
- OutOfMemoryError
- Memory Leak
- Distributing Java Code with JARs

Module 21: Overview of Java SE Apis

- Java GUI Programming
- Networking
- Security
- Databases - JDBC
- Concurrent Programming
- Naming - JNDI
- Management - JMX
- XML
- Web Services
- Remote Method Invocation
- Image I/O
- Printing
- Summary

Module 22: Overview of Java EE

- Goals of Enterprise Applications
- What is Java?
- What is Java EE?
- The Java EE Specifications
- Versions
- Role of Application Server
- Java EE Components
- What is a Servlet?
- Servlet Execution
- What is a JSP?
- JSP Code Sample
- Introduction to JSF
- Example JSF Page
- What is an EJB?
- EJB Types
- Java Persistence API
- EJB Examples
- Web Services
- Web Browser
- Other Clients
- Model-View-Controller Architecture
- MVC – An Example
- Java EE Vendor Specifications
- Containers
- Java EE Blueprint
- Java EE Application Structure
- EAR File
- What are Modules?
- Summary