Oracle 10*g*: Java Programming

Duration 5 Days

What you will learn:

This course teaches the concepts and essential elements of the Java language. While learning Java, you will build applications using standard Java technologies. The development environment used for this course is Oracle JDeveloper 10g. In addition to coding in Java, you will learn how to use the graphical development capabilities of JDeveloper to aid application development.

Audience:

J2EE Developer
Java Developer

Required Prerequisites:

Experience using a structured 3GL, such as C or Pascal

Suggested Prerequisites:

Basic knowledge of Object Oriented principles

Course Objectives:

Write standalone applications with the Java programming language
Build, generate, and test application components using Oracle JDeveloper 10g
Access database records using JDBC

Course Topics:

Introducing the Java and Oracle Platforms

What is Java?
Benefits of the Java Programming and Runtime environment
Object Oriented approach
Architecture Neutral deployment
Java and Internet computing
The Java Virtual Machine (JVM)
Oracle10g, OracleAS10g and Java

Defining Object Oriented Principles

Overview of classes, objects and methods
Communication between objects
Describe Abstraction, Inheritance, Encapsulation, Aggregation and Polymorphism
Private and public private keywords and encapsulation
Compare and Contrast Sub-classes and Class members
Examine the course application class model

Basic Java Syntax and Coding Conventions

Java Keywords
Java file structure (package, import, class)
Specifying Methods
Statements
Code Blocks
Java compiler

Exploring Primitive Data Types and Operators

Primitive data types
Declaring and initializing variables
Variable naming rules and conventions
Character literals and Unicode escape sequences
Boolean & Primitive variables
Assignment operators

Compound assignment operators
String object literals and the Concatenation Operator

Controlling Program Flow

Decisions: if, switch, conditional operator
Repetition: while, do-while, for
Break and continue
Enumerators and iterators
Labeled break and continue

Building Java Using Oracle JDeveloper

JDeveloper components
Workspaces and using the Project Wizard
Creating an Application
Building and running the application
Navigating in the Structure pane
Syntax highlighting and parameter matching
Modifying environment options

Creating Classes and Objects

Classes
Defining instance variables and instance methods
Creating objects using new and the default Constructor
Primitive variables vs. object references
Packages
Encapsulation using public and private access modifiers
Examples of class methods in the Java Library

Class Loading, Object Life Cycle and Inner Classes

Instance variables and instance methods: a review
Method overloading
Initializing instance variables using initializers
Initializing instance variables using constructors
Overloaded constructors
Defining Inner and Anonymous Classes
The finalize method

Using Strings, StringBuffer, Wrapper and formatting Classes

The Java String class
String conversion and comparison
Overview of the wrapper classes
Conversions to and from primitive types using the wrapper classes
DataFormat, DecimalFormat, and MessageFormat classes
Standard OutputStreams and PrintStreams
Writing and Reading Streams
Object Serialization

Reusing Code with Inheritance and Polymorphism

Subclassing and inheritance: the concepts
Inheritance hierarchies
Constructors and inheritance
Overriding superclass methods
Calling superclass methods
Using the instanceof operator
Class casting
Protected variables and methods

Using Arrays and Collections

Construct an array of primitives
Construct an array of object references
Process command line arguments
Handle command line parameters in a Java application
Using Vectors
HashTables and Properties

Reading command line and system, properties
Manually synchronizing ArrayLists and HashMaps

Structuring Code Using Abstract Classes and Interfaces

Abstract classes: the concepts and the syntax
Abstract methods
Defining and implementing interfaces
Polymorphism with abstract classes
Using interfaces to avoid multiple inheritance
Polymorphism with interfaces

Throwing and Catching Exceptions

What is an exception?
Throwable classes
Catching exceptions using try and catch
Ensuring code is executed using a finally block
Declared exceptions

User Interface Design: Planning a Form Layout

Brief History and comparison of AWT and Swing
Swing containers
Container Hierarchy to control component groups and layout
Adding a JButton and JTextfield components into a container
Building a GUI framework manually (practice w/o JDeveloper)
Using JDeveloper to build the GUI framework
The Swing container toolbar

Adding User Interface Components and Event Handling

Standard Swing components
Pluggable Look and Feel, and UIManager basics
The Java event model
Adding event handlers using Oracle JDeveloper
Model View Controller principles using a List component
Using JOptionPane for informational and error messages

Accessing the Database with JDBC

Java in the Database, the OracleJVM
Steps for using JDBC to execute a SQL statement
Registering the driver
Getting a database connection
Executing a SQL statement
Handling exceptions
Managing transactions

Deploying Applications Using Java Web Start

Architecture overview
Create a Web Start profile
Integrate business component
Deploy archive
Java Network Launching Protocol (JNLP)
Java Web Start with JClient
Files added to projects for Web Start

Suggested Next Courses:

Oracle JDeveloper 10*g*: Build Applications with ADF
Oracle 10*g*: Build J2EE Applications