

Oracle Forms Developer 10g

Length: 5 Days

Audience: Analysts and Application Developers responsible for the development or maintenance of application systems which access an Oracle database.

Prerequisites: Oracle SQL & SQL*Plus and **Oracle PL/SQL Programming** and familiarity with principles of Graphical User Interface (GUI) design.

Overview: In this course students build, test, debug, and deploy interactive Internet applications using Oracle Forms Developer 10g. Through lectures, demonstrations, and workshops, students will learn how to perform the following:

Customize forms with user input items such as check boxes, list items, radio groups, and Pluggable Java Components

Integrate Java into Forms applications by using JavaBeans

Control navigation, data access, validation, and transactions through event-related triggers

Display Forms elements in multiple windows using a variety of layout styles

Implement triggers

Reuse objects and code

Link one form module to another

Topics discussed include:

- Introducing Oracle Forms Developer and Forms Services
 - Oracle Application Server 10g Architecture
 - Grid Computing
 - Running a Forms Developer Application
 - Working in the Forms Developer Environment
- Creating Forms Modules
 - Creating a Basic Forms Module
 - Creating a Master-Detail Forms Module
 - Modifying the Data Block
 - Modifying the Layout
- Working with Data Blocks and Frames
 - Defining Blocks and Setting Properties
 - Using the Property Palette
 - Creating and Using Visual Attributes and Property Classes

- Creating Control Blocks
- Working with Input Items
 - Creating Text Items
 - Creating LOVs
 - Creating Check Boxes
 - Creating List Items
 - Creating Radio Groups
- Working with Non Input Items
 - Creating a Display Item
 - Creating an Image Item
 - Creating a Push Button
 - Creating a Calculated Item
 - Creating a Bean Area Item
- Working with Windows and Canvases
 - Overview of Windows and Canvases
 - Displaying a Forms Module in Multiple Windows
 - Creating a New Window
 - Displaying a Forms Module on Multiple Layouts
 - Creating a New Content Canvas
 - Creating a New Stacked Canvas
- Producing Triggers
 - Defining Trigger Components: Type, Code, and Scope
 - Using the PL/SQL Editor
 - Writing Trigger Code
 - Using Variables and Built-ins
- Debugging Triggers
 - The Debugging Process
 - The Debug Console
 - Setting Breakpoints
 - Running a Form in Debug Mode
 - Stepping through Code
- Adding Functionality to Items
 - Coding Item Interaction Triggers
 - Defining Functionality for Check Boxes
 - Changing List Items at Run Times
 - Displaying LOVs from Buttons
 - Interacting with JavaBeans
- Run-Time Messages and Alerts
 - Built-ins and Handling Errors
 - Controlling System Messages
 - The FORM-TRIGGER_FAILURE Exception
 - Using Triggers to Intercept System Messages
 - Creating and Controlling Alerts
 - Handling Server Errors
- Query Triggers
 - SELECT Statements Issued During Query Processing

- WHERE and ORDER BY Clauses
- Writing Query Triggers
- Query Array Processing
- Coding Triggers for Enter-Query Mode
- Overriding Default Query Processing
- Obtaining Query Information at Run Time
- Validation
 - Validation Process
 - Controlling Validation Using Properties
 - Controlling Validation Using Triggers
 - Tracking Validation Status
 - Using Built-ins to Control When Validation Occurs
- Navigation
 - Navigation Overview
 - Understanding Internal Navigation
 - Using Object Properties to Control Navigation
 - Writing Navigation Triggers
 - Using Navigation Built-ins in Triggers
- Transaction Processing
 - The Commit Sequence of Events
 - Testing the Results of Trigger DML
 - DML Statements Issued During Commit Processing
 - Overriding Default Transaction Processing
 - Getting and Setting the Commit Status
 - Implementing Array DML
- Sharing Objects and Code
 - Working with Object Libraries
 - Working with SmartClasses
 - Working with PL/SQL Libraries
 - Introducing Multiple Form Applications
 - Multiple Form Applications Overview
 - Starting Another Forms Module
 - Sharing Data Among Modules