

Oracle Forms Developer 11g

Duration: 5 Days

Overview: In this course students build, test, debug, and deploy interactive Internet applications using Oracle Forms Developer 11g. 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 and data in multiple canvases and windows
- Create form modules
- Display form modules in multiple windows using a variety of layout styles
- Implement triggers
- Reuse objects and code
- Link one form module to another

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

Topics Discussed Include:

Introducing Oracle Forms Builder and Forms Services

- Oracle Forms Services Architecture
- Running a Forms Application
- Working in the Forms Builder 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
- Creating and Using 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 Time
- 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

Using WebUtil to Interact with the Client

- Benefits of WebUtil
- Integrating WebUtil into a Form
- Interacting with the Client