

ITSE 1074

Course Syllabus

1. Name of Course: Build Internet Applications

2. Number of Clock Hours: 60 hours

3. Course Description:

Learn about Oracle's rapid application development tool for interactive Internet applications, featuring rich, desktop-style, user interfaces. Create a single forms module and deploy it on the web or client/server with no changes to the module. This is the development tool used by Oracle Applications. Learn how to implement it within your enterprise. In this course students build and test interactive Internet applications. Working in a graphical user interface (GUI) environment, students learn to customize Forms with user input items such as check boxes, list items, and radio groups. They also learn to modify data access by creating event-related triggers. Prepares students for the Oracle Certified Professional exam #1Z0-141.

4. Course Objectives

At the end of this course, students will be able to:

- Deploy Form modules on the Web
- Display Form modules in multiple windows and use a variety of layout styles
- Link one form module to another
- Reuse objects and code
- Create Form modules, including components for database interaction and GUI controls

5. Rationale:

Upon completion of this course, students will have a better understanding of Microsoft Oracle 9i, a powerful and popular database system.

6. Required Materials:

Oracle Official Curriculum included.

7. Evaluation

Those who participate in class discussions, complete course lab work, and miss no more than three class meetings will be awarded 6.0 continuing education units.

8. Course Outline

Introduction to Oracle Forms Developer and Oracle Forms Services

- Features and Benefits of Oracle Forms Developer and Oracle Forms Services
- Navigating Around the Form Builder Interface
- Using the Online Help Facilities
- The Course Application

Running a Form Builder Application

- The Runtime Environment
- Navigating a Forms Application
- Retrieving Both Restricted and Unrestricted Data from the Database into a Forms Application
- Identifying the Two Modes of Operation
- Inserting, Updating, and Deleting Records

- Displaying Database Errors

Working in the Form Builder Environment

- Identifying the Main Forms Executables
- Identifying the Main Components of Form Builder
- Identifying the Main Objects in a Form

Creating a Basic Form Module

- Creating a Form Module
- Creating a Data Block
- Modifying the Layout
- Saving and Running a Form Module
- Identifying Form File Formats and Their Characteristics
- Creating Data Blocks with Relationships
- Running a Master-Detail Form Module

Working with Data Blocks and Frames

- Managing Object Properties
- Controlling the Behavior and Appearance of Data Blocks
- Creating Data Blocks that Do Not Directly Correspond to Database Tables
- Deleting Redundant Data Blocks and Their Components

Working with Text Items

- Describing Text Items
- Creating a Text Item
- Modifying the Appearance of a Text Item
- Controlling the Data in a Text Item
- Altering the Navigational Behavior of a Text Item
- Enhancing the Relationship Between the Text Item and the Database
- Adding Functionality to a Text Item
- Including Helpful Messages

Creating Lists of Values (LOVs) and Editors

- Describing LOVs and Editors
- Designing, Creating, and Associating LOVs with Text Items
- Creating an LOV Using the LOV Wizard
- Creating Editors and Associating Them with Text Items

Creating Additional Input Items

- Identifying the Item Types that Allow Input
- Creating a Check Box
- Creating a List Item
- Creating a Radio Group

Creating Noninput Items

- Identifying Item Types that Do Not Allow Input
- Creating a Display Item
- Creating an Image Item
- Creating a Sound Item
- Creating a Button
- Including Tooltips
- Creating a Calculated Field
- Including a Hierarchical Tree Control

Creating Windows and Content Canvases

- Describing the Relationship Between Windows and Content Canvas-Views
- Displaying a Form Module in Multiple Windows
- Displaying a Form Module on Multiple Layouts

Working with Other Canvas-Views

- Describing the Different Types of Canvas-View
- Identifying the Appropriate Canvas-View Type for Different Scenarios
- Creating an Overlay Effect on the Interface
- Creating a Toolbar
- Creating a Tabbed Interface

Introduction to Triggers

- Identifying the Different Trigger Categories
- Planning the Type and Scope of Triggers in a Form
- Describing the Properties that Affect Trigger Behavior

Producing Triggers

- Writing Trigger Code
- Explaining the Use of Built-In Subprograms in Oracle Developer Applications
- Identifying the When-Button-Pressed Trigger and Knowing When to Use It
- Identifying the When-Window-Closed Trigger and Knowing When to Use It
- Debugging Triggers and Program Units

Adding Functionality to GUI Items

- Supplementing the Functionality of Input Items Using Triggers
- Supplementing the Functionality of Noninput Items Using Triggers
- Handling Errors Using Built-In Subprograms

Runform Messages and Alerts

- Describing the Default Messaging Behavior of a Form
- Identifying the Different Types of Forms Messages
- Controlling System Messages
- Creating and Controlling Alerts

Query Triggers

- Explaining the Processes Involved in Querying a Data Block
- Identifying Where Triggers Can Be Coded to Control the Query Process
- Writing Triggers that Screen Query Conditions
- Writing Triggers to Supplement Query Results
- Controlling Trigger Action Based Upon the Query Status of the Form

Validation

- The Effects of the Validation Unit upon a Form
- Employing Additional Forms Features for Validation
- Controlling Validation Using Triggers
- Supplementing Default Validation

Navigation

- Distinguishing Between Internal and External Navigation
- Describing and Using the Navigation Triggers
- Writing Navigation in Triggers

Transaction Processing

- Explaining the Process Used by Oracle Forms Developer to Apply Users' Changes to the Database
- Describing the Commit Sequence of Events
- Supplementing Transaction Processing
- Allocating Sequence Numbers to Records as They Are Applied to Tables

Writing Flexible Code

- Defining Flexible Code
- The Advantages of Using System Variables
- Identifying Built-In Subprograms that Assist Flexible Coding
- Referencing Objects by Internal ID
- Referencing Items Indirectly

