

## **JavaServer Faces Development with IceFaces**

### **Course Summary**

#### **Description**

This class covers how to develop and write Web applications that use the JavaServer Faces (JSF) framework. Standard JSF and IceFaces components are covered.

#### **Topics**

- Introduction to the Workshop
- JSF Overview
- Ajax Overview
- JSF Components
- Validating Input
- JSF Navigation
- Internationalization
- Writing Custom Components
- Advanced JSF and IceFaces Topics

#### **Audience**

This course is developed for experienced Java Web developers that wish to learn how to use JSF and IceFaces.

#### **Prerequisites**

Before taking this course students should have prior Java JSP and servlet programming experience.

#### **Duration**

Three days

## JavaServer Faces Development with IceFaces

### Course Outline

- I. Introduction to the Workshop**
  - A. J2EE Web Application Review
  - B. Servlet Review
  - C. JSP Review
  - D. Model 2 Architectures
- II. JSF Overview**
  - A. What is JSF?
  - B. JSF Architectures
  - C. JSF versus Struts and Other Frameworks
  - D. JSF Requirements
- III. Ajax Overview**
  - A. What is Ajax
  - B. Using XMLHttpRequest
  - C. Overview of Ajax Libraries
  - D. Introduction to IceFaces
- IV. JSF Components**
  - A. What are JSF Components?
  - B. Using Standard JSF Components
  - C. Using IceFaces Components
    - 1. Common Attributes
    - 2. Using IceFaces componets
    - 3. Enhanced standard components.
- V. Validating Input**
  - A. Validation Scenarios
  - B. Using Standard Validators
  - C. Writing Custom Validators
  - D. Using partial submit
- VI. JSF Navigation**
  - A. JSF Event Model
  - B. Navigating Between Pages
- VII. Internationalization**
  - A. Message Bundles and Resource Files
  - B. Localizing Output
- VIII. Writing Custom Components**
  - A. Extending an Existing Component
  - B. Writing a New Component
- IX. Advanced JSF and IceFaces Topics**
  - A. Connection management
  - B. Drag and drop features
  - C. Adding and customizing effects
  - D. Using Facelets