Quiz Test for Chapter 5(JSF) and 6(Simple Tags)

Time: 30 Minutes

**Descriptive question on JSF**

1. **What are the advantages of JSF?**

**Ans:** The JSF specification lists the following ways that JSF helps web-application developers to create user interfaces (UIs):

• Makes it easy to construct a UI from a set of reusable UI components

• Simplifies migration of application data to and from the UI

• Helps manage UI state across server requests

• Provides a simple model for wiring client-generated events to server-side application code

• Allows custom UI components to be easily built and reused

# What are tags in JSF?

# Ans:

# 

# What is JSF life cycle and its phases?

# Ans:

1. **What is the type of request /response handle by JSF?**

**Ans:** We are concerned with these three request/response pairs:

• Non-JSF request generates JSF response

• JSF request generates JSF response

• JSF request generates non-JSF response

1. **Write the name of JSF libraries?**

**Ans:**

• Six JSF JARs: commons-beanutils.jar, commons-collections.jar, commons-digester.jar,   
commons-logging.jar, jsf-api.jar, and jsf-impl.jar

• Two JSTL JARs: jstl.jar and standard.jar

1. **What is the job of faces-config.xml class?**

**Ans:** Information about the view components in the web application and information about how control flows through the application is contained in a special configuration file named faces-config.xml.

1. **Write the event handling process in JSF?**

**Ans:** Value Change Listeners

Action Listeners

Calling Multiple Listeners

1. **What do you mean by MVC in JSF?**

**Ans:** The primary design pattern of JSF is the Model-View-Controller (MVC) pattern. MVC separates an application architecture into three categories of components: model, view, and controller. The model is the abstraction of all the domain data in the system. The view is the visualization of the model. In a web application, the view consists of the HTML pages and the components that create the HTML pages sent to web browsers, the Wireless Application Protocol (WAP) pages sent to mobile devices, or the UI components sent to a dedicated client. The controller is the set of components that manage the communications between model and view.

1. **Need for Custom Tags**

**Ans:** Among the best practices that have been established around JSP, one of the most important suggests that there should be as little Java code embedded inside a JSP page as possible. Experience has shown us that three key factors benefit from this practice:

• Reusability

• Readability

• Maintainability

1. **What is Tag File**

**Ans:** In short, a tag file is simply a JSP fragment containing some content or JSP code that you would like to use over and over again. This fragment is accessed by using a custom tag.

1. **Body content of tag file**

**Ans:** Body content is defined as anything that falls between the start and end tags. When first introduced, three types of body content were defined: *empty, tagdependent, and JSP*. JSP 2.1 supports the same three categories of body content, and an additional category named *.*

1. **Difference between JavaBeans vs. Custom Tags**

**Ans:**

• Use JavaBeans for representing and storing information and state. An example is building JavaBeans to represent the business objects in your application.

• Use custom tags to represent and implement actions that occur on those JavaBeans, as well as logic related to the presentation of information. An example from JSTL is iterating over a collection of objects or conditional logic.

1. **Difference between Simple tag vs. Custom tag**

**Ans:** To build classic tags, you write the functionality provided by the custom tag as a Java class that implements the javax.servlet.jsp.tagext.Tag interface.

Two new ways for building custom tags have been introduced: tag files and simple tags. Tag files and simple tags allow the functionality of custom tags to be implemented by using JSP fragments and Java code, respectively. Although they use different paradigms, they both greatly simplify the way in which custom tags can be built and have changed the way that JSP-based web applications are built.

1. **Define Simple Tag Interface**

**Ans:** The SimpleTag interface defines the basic contract between simple tags and the JSP page on which they are used. The interface itself serves two purposes:

• It provides the simple tag with information about its execution environment.

• It provides a method for executing the functionality encapsulated by the simple tag handler.

1. **Request time expression**

**Ans:**

1. **What is TLD? How you configure a tag handler?**

**Ans:**