Assignment-2

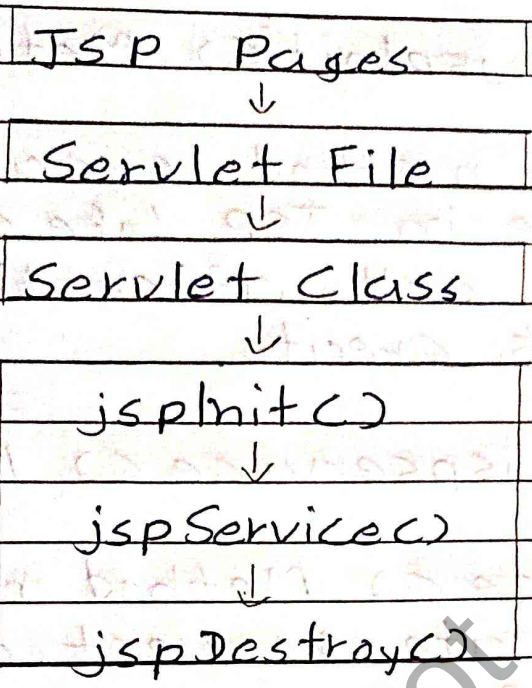
* Explain JSP with Its Life Cycle.

=> JSP is stands for Java Server Page which is used to create web application like servlet

JSP is the extension of Java servlet Technology.

JSP Pages are contains the following steps.

(i) Translation of JSP Pages
(ii) Compilation of JSP Pages
(iii) Loads class file
(iv) Servlet Object is create
(v) Invokes jspInit() method
(vi) Invokes jspService() method
(vii) Invokes jspDestroy() method.



(i) Translation of JSP Pages: In this phase, we have to create JSP Page.

(ii) Compilation of JSP Pages: In this phase, we have to create servlet Page.

(iii) Loads class File: In this phase, we have to load java class File into memory.

(iv) Servlet Object is Create: In this phase, object of class is create.

(v) Invokes jspInit() Method :

jspInit() method is called only one time in JSP Life Cycle which is call after the servlet Object is Create.

(vi) Invokes jspservice() Method :

jspService() Method is used to generate request and response in JSP Pages.

(vii) Invokes jspDestroy() Method :

jspDestroy() method is called only one time after the complicition of request and response in JSP Pages.

* Explain JSP Implicit Objects.

=> There are 9 types of JSP Implicit Objects.

1 out Objects:

This Object is used to write

the any data in the JSP Page.

~~Syntax :~~

Ex:

```
<%.
    out.print("Welcome Brain
        Spot");
%.>
```

2. Request Objects:

This Object is to do request in the any JSP Page.

~~Syntax :~~

Ex:

```
<%.
    String a = request.getParameter
        ("Brain Spot");
    out.print(a);
%.>
```

3 Response Objects:

This Object is used to get response in the any JSP Pages.

Page No.
Date : / /

Ex :

```
<%
    response.sendRedirect
      ("https://thebrainspot.in");
%>
```

4 Config Object :

This Object is used to initialization parameter for a any JSP Page.

Ex.

```
<%
    String name = config.
      getInitParameter("Brain
      Spot");
%>
```

5 Application Object :

This Object is used to initialization parameter from configuration file.

Ex.

```
<%
    String name = application.
      getInitParameter("Brain
      Spot");
```

-%. >

6 Session Object :

This Object is used to set, get or remove attribute in session information.

Ex :

```
<-%.
    String a = request.getParameter
            ("Brain Spot");

    Session.setAttribute(a);
-%. >
```

7 Page Context Object :

This Object is used to set, get or remove attribute From Page.

Ex.

```
<-%.
    String a = request.getParameter
            ("Brain Spot");

    pageContext.setAttribute(a);
-%.>
```

Page No.
Date : / /

8 Page Object :

This Object is used to assigned to the reference of servlet class.

9 Exception Object :

This Object is used to print the exception in error page.

Ex.

```
<%= exception %>
```

* Difference between JSP and servlet.

=>	JSP	Servlet
1	JSP is a HTML-based compilation code.	Servlet is a Java code.
2	JSP only accepts HTTP Requests.	Servlet can accepts all the Protocol requests.
3	service() method can not Override.	Service() method can be override

4	JSP have Implicit Object.	Servlet does not have inbuilt Implicit Object.
5	JSP have Custom Tags.	Servlet does not have custom tags.
6	JSP acts as a Viewer.	Servlet acts as a Controller.

* Explain JSP Processing ~~Life~~ Life Cycle.

=> JSP Processing Life cycle gives step by step Process of JSP Requests.

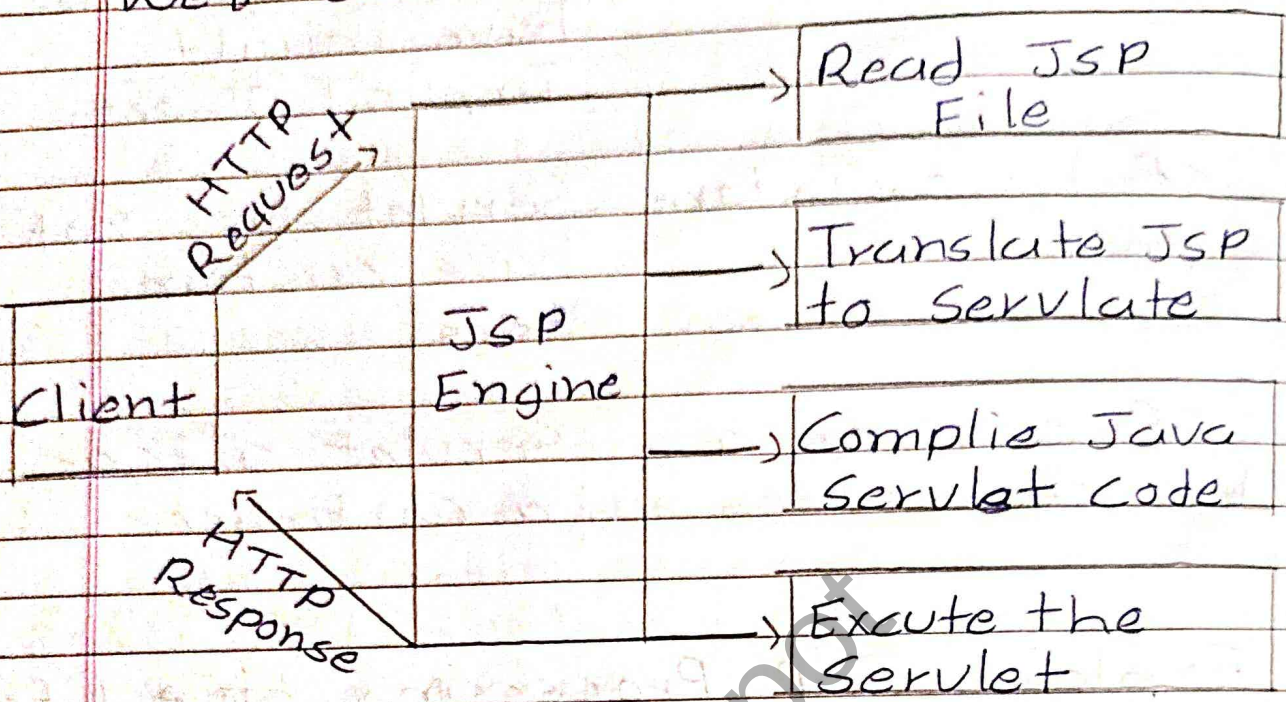
Using JSP Processing Life cycle we can easily understand how the JSP Request is Process.

This are the Basic steps of JSP Life cycle Process.

1 The Client is create a .jsp extension file and Using HTTP requests client request to

Page No.
Date : / /

Web browser.



2 Web ~~Serv~~ Browser request is forwarded to the JSP Engine.

3 The JSP Engine loads the JSP File and Translate JSP to servlet File.

4 The JSP Engine complie the servlet into .class file.

5 .class file is executed by the servlet engine.

6 The Web Browser forwards the HTML File to the Client's browser.

* Explain Various JSTL XML Tags.

=> Using JSTL XML Tags, we can create the XML Documents.

The JSTL XML tag libary use custom tags for interacting with XML data.

For Use XML tag we have to add one URL in JSP.

Syntax:

<%@ taglib Uri = "http://java.sun.com/jsp/jstl/xml" %>

This are the Basic JSTL XML Tags.

a. X:out - Used For XPath expressions.

b. X:parse - Used for specified XML data in the body tag.

c. X:set - Used to set a value of the XPath expressions.

d x:Choose - Used to Provides conditional operations.

e x:when - Used in body for condition is evaluated be 'true'.

f x:otherwise - Used in body for condition is evaluated be 'false'.

g x:if - Used for evaluating the test XPath expression if it is true than it will process in body content.

h x:transform - Used in a XML document for Provides the XSL.

i x:param - Used for set the parameter in XSLT style sheet.

Example:

```
<%@ taglib prefix="c" uri=
    "http://java.sun.com/jsp/jstl
     /core" %>
```

```
<%@ taglib prefix="x" uri="http://
    java.sun.com/jsp/jstl/xml" %>

<html>
  <body>
    <c:set var="V">
      <Vs>
        <VJ>
          <n> A </n>
        </VJ>
      </Vs>
    </c:set>
    <x:parse xml="${V}" vat=
      "Output" />
    <x:out select="$output /Vs
      /VJ/n" />
  </body>
</html>
```


* Explain JSTL tags with example.

=> JSP Standard Tag Library is used to represents the set of tag with different types.

There are Five types of JSTL tages.

(i) Core Tags
(ii) Function Tags
(iii) Formatting Tags
(iv) XML tags
(v) SQL tags.

(i) Core Tags:

The JSTL Core Tag is provides variable support, URL Management etc in JSP.

For Core Tag we have to use one url: http://java.sun.com/jsp/jstl/core

In Core Tag, we have to use Prefix as a c.

(ii) Function Tags:

The JSTL Function Tag is provides string related function.

For Function Tag, we have to use one url: http://java.sun.com/jsp/jstl/Functions.

In Function tag, we have to use Prefix as a Fn.

(iii) Formatting Tag:

The JSTL formatting tag is provides support for message formatting, number formatting or data Fomatting in JSP.

For ~~Fun~~ Formatting Tag, we have to use one url: 'http://java.sun.com/jsp/jstl/fmt'

In Formatting Tag, we have to use Prefix as a fmt.

(iv) XML Tag:

The JSTL XML tag is provides flow control, transformation etc in JSP.

For XML Tag, we have to use one url: 'http://java.sun.com/jsp/jstl/x'

In XML tag, we have to use Prefix as a x.

(v) SQL Tags:

The JSTL SQL Tag is used to Provides SQL support in JSP.

For SQL Tag, we have to use one url : 'http://java.sun.com/jsp/jstl/xml.

In SQL Tag, we have to use Prefix as a sql.

* Explain JSP Custom Tag.

=> JSP Custom tags are user defined tags.

Using Custom tag, we can separate the business Logic from the JSP Page.

When we use Custom tag then we no need to use Scriptlet tag.

After the Separate Logic we can easily manage the JSP Page.

There are two ~~ta~~ Method to Create Custom Tag.

(i)

<prefix:tagname attribute=values />

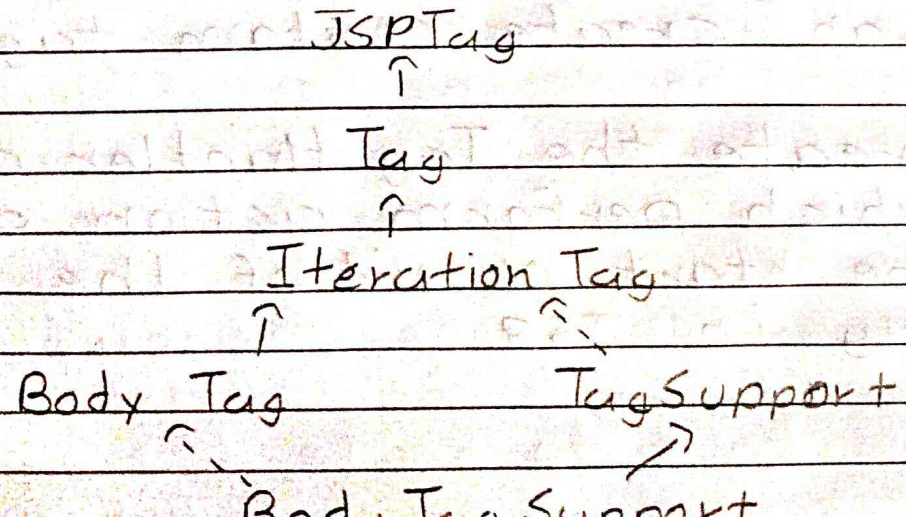
(ii) <prefix:tagname attributes=values

// body code

</prefix:tagname>

Custom Tag is also Provides JSP Custom tag API Support.

For JSP Custom tag API, we have to use javax.servlet.jsp.tagext package.



Jsptag Interface is a root interface that consists all the interfaces and classes.

Tag Interfaces is sub interfaces of Jsptag interface which provides methods to perform action.

IterationTag Interface is the sub interfaces of the tag interface which provides extra method for body.

TagSupport class implements the iterationtag interface which is provides new base class for Tag Handlers.

Using this three steps, we can create custom tag.

1 Create the Tag Handler class which perform action at the start or end of the tag in JSP.

2 Create the Tag Libary Descriptor File which is contains tag related information.

3 Create JSP File and uses the custom tag to write Business Logic in JSP.

* Explain Exception Handling in JSP.

=> Exception Handling is used to handles the runtime error in JSP Program.

There are two ways to Perform Exception Handling in JSP.

1) By Error Page and isErrorPage attributes of Page directive

2) By <error-page> elements in web.xml File.

1 By Error Page and is ErrorPage attributes of Page Directive.

In this method, we have to use Page directive as isErrorPage.

We have to create a one error handler File.

Example:

<%@ page isErrorPag = "true" %>

<h1> Error </h1>

Exception is : <%= exception %>

This Page can display the error or exception message.

2 By <error-page> element in web.xml File.

In this method, we do not need to define errorPage attribute for each jsp page.

We have to write Only once for error message.

We have to write the error message in web.xml File.

Example:

```
<web-app>

  <error-page>

    <exception-type> java.lang.
        Exception
    </exception-type>
    <location> / error.jsp </location>

  </error-page>

</web-app>
```

Page No.
Date : / /

* Explain JSP Action Tag.

=> JSP Action Tag is used to Perform the specific tasks in the JSP File.

Using Action Tag, we can control the flows between the JSP Pages and Java Bean.

This are the Basic JSP Action Tag.

- jsp:forward - This tag is used to forward the request and response to other resource.

- jsp:include - This tag is used tho include other resource in jsp file.

- jsp:useBean - This tag is used to create or locate the bean object in JSP.

- jsp:setProperty : This tag is used to set the value of the bean object in JSP.

- jsp:getProperty : This tag is used to print the value of bean Object in JSP.

- jsp:plugin : This tag is used to add the plugin in JSP File.

- jsp:param : This tag is used to set Parameter value in JSP File.

- jsp:fallback : This tag is used to print plugin working or not message in JSP.

Example:

```
<html>
  <title> JSP </title>

  <body>

      <jsp:include page = "java.jsp"
          flush = "true" />
  </body>

</html>
```