

## ASSIGNMENT 2

NAME: SUSNATO BARUA

ROLL NO: 1854044

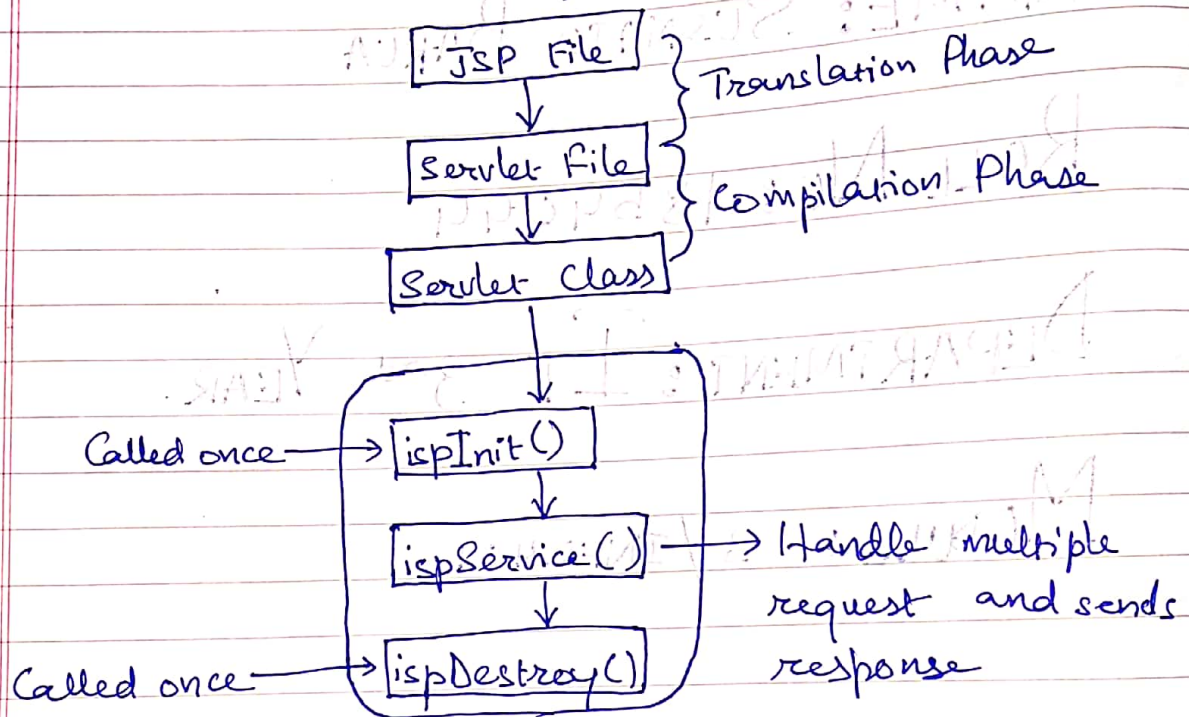
DEPARTMENT: IT 3<sup>rd</sup> YEAR.

MODULE 3-1 3 / INFO3101

1. Draw and explain JSP lifecycle.

Ans: A Java Server Page life cycle is defined as the process started with its creation which later translated to a servlet life cycle comes into play. This is how the process goes on until its destruction.

### JSP Life Cycle



Following steps are involved in JSP life cycle:-

- Translation of JSP page to Servlet.
- Compilation of JSP page (Compilation of JSP into test.java)
- Classloading (test.java to test.class)
- Instantiation (~~Off~~ Object of the generated Servlet is created)
- ~~Instant~~ Initialization (jspInit() method is invoked by the container)
- Request processing (jspService() is invoked by the container)
- JSP cleanup (jspDestroy() method is invoked)



by the container.)

### Translation of JSP page to Servlet:

This is the first step of JSP life cycle. This translation phase deals with Syntactic correctness of JSP. Here test.jsp file is translated to test.java.

### Compilation of JSP page:

Here the generated java servlet file (test.java) is compiled to a class file (test.class).

Classloading: Servlet class which has been loaded from JSP source is now loaded into container.

Instantiation: Here instance of the class is generated. The container manages one or more instance by providing response to requests.

Initialization: `jspInit()` method is called only during the life cycle immediately after the generation of servlet instance from JSP.

Request processing: `jspService()` method is used to serve the raised requests by JSP. It takes request and response object as parameters. This method cannot be overridden.

JSP Cleanup: In order to remove the JSP from use by the container or to destroy method for servlets `jspDestroy()` method is used. This method is called once, if you need to perform



any cleanup task like closing open files, releasing database connections `jspDestroy()` can be overridden.

2. Highlight the differences between servlet and JSP.

Ans. Servlet

JSP

a) Servlet is a java code.

a) JSP is an <sup>an</sup> html based code.

b) Writing code for servlet is harder than JSP as it is ~~is~~ ~~JSP~~ html in java.

b) JSP is easy to code as it is java in html.

c) Servlet plays a controller role in MVC approach.

c) JSP is the view in MVC approach for showing output.

d) Servlet is faster than JSP.

d) JSP is slower than Servlet because the first step in JSP lifecycle is the translation of JSP to java code and then compile.

e) Servlet can accept all protocols requests.

e) JSP only accepts http requests.

f) In servlet, we can override the `service()` method.

f) In JSP, we cannot override its `service()` method.

3. What is taglib directive in JSP and how it is used?

Ans: The taglib directive declares that your JSP page uses a set of custom tags, identifies the location of the library, and provides means for identifying the custom tags in your JSP page. It is used to define a tag library that defines many tags.  
Syntax:-

<%@ taglib uri = "uri of the tag library"  
prefix = "prefix of tag library" %>



4. Write a JSP program to develop library management system. Write all necessary assumption.

5. Write a JSP scriptlet to calculate the factorial of a number.

Ans:-

<HTML>

<HEAD>

<TITLE> Recursion </TITLE>

</HEAD>

<BODY>

<H1> Factorial number using recursion </H1>

<%!

int numberfactorial (int number)

{

if (number == 1)

{

return number; }

else {

return number \* numberfactorial (number - 1);

}

}

}



```

    <%
    out.println ("The factorial of 5 is" +
    numberfactorial (5));
    %>
</BODY>
</HTML>

```

OUTPUT:- The factorial of 5 is 120

★ Q3- Example:-

```

<html>
<body>
<%@ taglib url = "http:// www.google.com/
tags" prefix = "mytag" %>
<mytag : current-Date >
</body>
</html>

```

Q6- With the help of an example explain JSP FORWARD ACTION

Ans- The forward action terminates the action of the current page and forwards the request to another resource such as a static page, another JSP page, or a Java servlet.

Example:- Let us reuse the following two files  
(a) date.jsp and main as follows:-

date.jsp

```

<p> today's date: <%= (new java.util.Date().
toLocal

```

main.jsp :-

```

<html>
<head>

```

```

<title> This includes action </title>

```

```
</head>
<body>
  <center>
    <h2> This includes action </h2>
    <jsp.forward page = "date.jsp" />
  </center>
</body>
</html>
```

Q7. Briefly explain the components of JDBC.

Ans:- To communicate with the Database using JDBC you need the following components.

JDBC Drive Manager: The Drive Manager class of the `java.sql` package manages different types of JDBC drivers. This class ~~to~~ loads the driver classes. In addition to this, whenever a new connection is established it chooses and loads the suitable ~~for~~ driver from previously loaded one.

JDBC API:- It is a Java abstraction which enables applications to communicate with relational databases. It provides two main packages namely `java.sql` and `javax.sql`.

JDBC ODBC Bridge Driver:- This is a bridge driver which translates the JDBC method calls to ODBC function calls using the package you can communicate with database which uses ODBC drivers.