Generic Servlets

Step 1: Creating a dynamic web project

- OpenEclipse
- Go the File menu. Choose New->Dynamic Web Project
- Enter the project name as ServletConcept. Click on Next
- Enter nothing in the next screen and click on Next
- Check the checkbox Generate web.xml deployment descriptor and click on Finish
- This will create the project files in the Project Explorer

Step 2: Creating an HTML page

- In the Project Explorer, expand the project 'ServletConcept'
- Expand WebContent. Right click on WebContent . Choose New->HTML File
- Enter the filename as index.html and click on Finish
- Enter the following code:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Name Servlet</title>
</head>
<body>
	<center>
```

```
        <form name=frmName method="GET" action="name">
            <table width="50%" border="0">
                <tr valign="top">
                    <td width="40%">Enter your First
Name</td>
                    <td><input name="fname" id="fname"
maxlength=30></td>
                </tr>
                <tr valign="top">
                    <td width="40%">Enter your Last
Name</td>
                    <td><input name="lname" id="lname"
maxlength=30></td>
                </tr>
                    <tr valign="top">
                    <td colspan=2 width="100%">
                        <button>Submit</button>
                    </td>

                </tr>
            </table>
        </form>
    </center>
</body>
</html>
```

- Click on the Save icon

Step 3: Creating a servlet

- In the Project Explorer, expand ServletConcept->Java Resources
- Right click on src and choose New->Servlet
- In Class Name, enter NameServlet and click on Finish
- Enter the following code:

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
* Servlet implementation class NameServlet
*/
@WebServlet("/NameServlet")
public class NameServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
* @see HttpServlet#HttpServlet()
*/
    public NameServlet() {
        super();
        // TODO Auto-generated constructor stub
```

```
  }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {
        // TODO Auto-generated method stub
        String fname = request.getParameter("fname");
        String lname = request.getParameter("lname");

        PrintWriter out = response.getWriter();
      out.println("<html><body>");
      out.println("Your full name is " + fname + " " + lname);
      out.println("</body></html>");
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
```

}

Step 4: Configuring web.xml

- In the Project Explorer, expand ServletConcept-> WebContent->WEB-INF
- Double click on web.xml to open it in the editor
- Enter the following script:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
  <display-name>ServletDemo</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>

  <servlet>
      <servlet-name>NameServlet</servlet-name>
      <servlet-class>NameServlet</servlet-class>
```

```
 </servlet>

 <servlet-mapping>
     <servlet-name>NameServlet</servlet-name>
     <url-pattern>/name</url-pattern>
  </servlet-mapping>

</web-app>
```

Step 5: Checking for servlet-api.jar

- Before building the project, we need to add servlet-api.jar to the project
- Servlet-api.jar file is already present in your practice lab. (Refer FSD: Lab Guide - Phase 2)
- To add it to the project, follow the below mentioned steps:
    - In the Project Explorer, right click on ServletConcept and choose Properties
    - Select Java Build Path from the options on the left
    - Click on Libraries tab on the right
    - Under ClassPath, expand the node that says Apache Tomcat
    - If there is an existing entry for the servlet-api.jar, then

click on Cancel and exit the window

- If it is not there, then click on Classpath entry and click on Add External JARs button on the right
- From the file list, select servlet-api.jar file and click on Ok
- Click on Apply and Close

Step 6: Building the project

- From the Project menu at the top, click on Build
- If any compile errors are shown, fix them as required

Step 7: Publishing and starting the project

- If you do not see the Servers tab near the bottom of the IDE, go to Window menu and click on Show View->Servers
- Right click on the Server entry and choose Add and Remove
- Click the Add button to move ServletConcept from the Available list to the Configured list
- Click on Finish
- Right click on the Server entry and click on Publish
- Right click the Server entry and click on Start
- This will start the server

Step 8: Running the project

- To run the project, open a web browser and type: http://localhost:8080/ServletConcept

Step 9: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master