**Rubric – CLC Mini-Project**

|  |  |  |
| --- | --- | --- |
| **Category** | **Comments** | **Score** |
| Explanation of the team’s solution to questions or challenges posed by mini-project. |  | /10% |
| Diagram, flowchart, or pseudo code . |  | /10% |
| Team and individual responsibilities are clearly articulated. |  | /10% |
| List of Java EE concepts, tools, and techniques used in this mini-project. |  | /10% |
| Correct use of English language and grammar in all project components. |  | /10% |
| Code submission to remote repository. |  | /10% |
| Code runs correctly and matches your flowchart/diagram. |  | /20% |
| Code Readability/Organization. |  | /10% |
| Specific requirements applying to this specific mini-project are implemented (i.e., Use of particular Java EE constructs, processes). |  | /10% |
| **TOTAL** |  | /100% |

Caleb Miller

20171119

CST-235

**CLC Mini-Project**

**Question 1**

How can EJB be used with JSP? Substantiate your answer with code examples and screenshots of execution. Seek instructor guidance as needed.

EJB’s can be used on many different platforms and when calling from a JSP, the server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. Below is a JSP calling from a EJP.

*<%*

*env.put(Context.INITIAL\_CONTEXT\_FACTORY,"weblogic.jndi.example");*

*env.put(Context.PROVIDER\_URL,"t3://localhost:8080");*

*String JNDI\_NAME = xxxx/MyBean;*

*javax.naming.Context initContext = null;*

*java.util.Hashtable env = new java.util.Hashtable(1);*

*try {*

*initContext = new javax.naming.InitialContext(env);*

*}*

*catch(Exception e) {*

*out.println("ERROR ");*

*}*

*Object beanObject = null;*

*try {*

*beanObject = initContext.lookup(JNDI\_NAME);*

*IMyBeanHome home = (MyBeanHome) beanObject;*

*IMyBean remote = home.create();*

*//Call EJB methods*

*double resultado = remote.calcBonus(3, 4.0);*

*}*

*catch(Exception e) {*

*out.println("ERROR " + JNDI\_NAME);*

*}*

*%>*

(Elias Manrique. 2000)

**Question 2**

How do EJBs enable portability and scalability of applications? Substantiate your answer with code examples and screenshots of execution. Seek instructor guidance as needed.

Enterprise JavaBeans enables rapid and simple development of distributed transactional secure and portable applications. The ease of portability with EJB’s means that the user gets to choose what implementation suits them best, this makes the application portable across EJB containers from different vendors.

*<html>*

*<head>*

*<title>Calculator</title>*

*</head>*

*<body bgcolor="lightgreen">*

*<h1>Basic Operations</h1>*

*<hr>*

*<form action="Result.jsp" method="POST">*

*<p>Enter first value:*

*<input type="text" name="num1" size="25"></p>*

*<br>*

*<p>Enter second value:*

*<input type="text" name="num2" size="25"></p>*

*<br>*

*<b>Select your choice:</b><br>*

*<input type="radio" name="group1" value ="add">Addition<br>*

*<input type="radio" name="group1" value ="sub">Subtraction<br>*

*<input type="radio" name="group1" value ="multi">Multiplication<br>*

*<input type="radio" name="group1" value ="div">Division<br>*

*<p>*

*<input type="submit" value="Submit">*

*<input type="reset" value="Reset">*

*</p>*

*</form>*

*</body>*

*</html>*

*</form>*

*<%@ page contentType="text/html; charset=UTF-8" %>*

*<%@ page import="com.javacodegeeks.example.ejb.\*, javax.naming.\*"%>*

*<%!*

*private OperationsSessionBeanRemote ops = null;*

*float result = 0;*

*public void jspInit() {*

*try {*

*InitialContext ic = new InitialContext();*

*ops = (OperationsSessionBeanRemote)ic.lookup(OperationsSessionBeanRemote.class.getName());*

*System.out.println("Loaded Calculator Bean");*

*} catch (Exception ex) {*

*System.out.println("Error:"*

*+ ex.getMessage());*

*}*

*}*

*public void jspDestroy() {*

*ops = null;*

*}*

*%>*

*<%*

*try {*

*String s1 = request.getParameter("num1");*

*String s2 = request.getParameter("num2");*

*String s3 = request.getParameter("group1");*

*System.out.println(s3);*

*if (s1 != null && s2 != null) {*

*Float num1 = new Float(s1);*

*Float num2 = new Float(s2);*

*if (s3.equals("add")) {*

*result = ops.add(num1.floatValue(), num2.floatValue());*

*} else if (s3.equals("sub")) {*

*result = ops.subtract(num1.floatValue(), num2.floatValue());*

*} else if (s3.equals("multi")) {*

*result = ops.mutliply(num1.floatValue(), num2.floatValue());*

*} else {*

*result = ops.divide(num1.floatValue(), num2.floatValue());*

*}*

*%>*

*<p>*

*<b>The result is:</b> <%= result%>*

*<p>*

*<%*

*}*

*}// end of try*

*catch (Exception e) {*

*e.printStackTrace();*

*//result = "Not valid";*

*}*

*%>*

(Dhruv Gupta. 2016)

**References**

Elias Manrique. 2000. EJB Calling From JSP. Retrieved from http://www.theserverside.com/discussions/thread/1726.html

What When How. N.D. EJB Overview. Retrieved from

http://what-when-how.com/enterprise-javabeans-3/ejb-overview/

Dhruv Gupta. 2016. Java Code Geeks. Retrieved from https://examples.javacodegeeks.com/enterprise-java/ejb3/ejb-tutorial-beginners-example/