

EJ75436SS Enterprise Java Solution Set November 2019

1.	Attempt <u>any three</u> of the following:
a.	What is Java Enterprise Edition (Java EE)? Explain. (Reference Book Page No.- 17) Diagram - 1 marks Explanation - 4 Marks
b.	Write a note on Multi Tire EE application architecture. (Reference Book Page No.- 59) Diagram - 1 marks Explanation - 4 Marks
c.	List the tasks that Servlet can do? (Reference Book Page No.- 83)
d.	Explain the life cycle of servlet. (Reference Book Page No.- 110) Diagram - 1 Marks Explanation of init() & destroy() - 2 Marks Explanation of service() - 2 Marks
e.	Write a servlet code to display Square and Square root of numbers between 25 and 365 in tabular form. import java.io.*; import javax.servlet.*; import javax.servlet.http.*; public class Test extends HttpServlet{ public void doGet(HttpServletRequest hreq, HttpServletResponse hres) throws ServletException, IOException{ PrintWriter pw = hres.getWriter(); out.println("<table border=1>"); for(int i=25; i<=365; i++){ out.println("<tr><td>"+(i*i)+"</td><td>"+Math.sqrt(i)+"</td></tr>"); } out.println("</table>"); } }
f.	List and explain four types of JDBC drivers. (Reference Book Page No.- 151) List 1 Marks Type -1 JDBC-ODBC Bridge Driver Type-2 Java Native Type-3 Net Protocol Driver Type-4 Pure Java Explanation With Diagram - 1 Marks Each
2.	Attempt <u>any three</u> of the following:
a.	Explain Cookie class with its constructor and any five methods. (Reference Book Page No.- 193) Description : 1 Marks Cookie() Cookie(String, String) 1 Marks Methods 3 Marks
b.	Write a servlet program to create a session and display the following: i) Session ID ii) New or Old iii) Creation Time (Reference Book Page No.- 240) import java.io.*; import javax.servlet.*; import javax.servlet.http.*; public class Test extends HttpServlet{ public void doGet(HttpServletRequest hreq, HttpServletResponse hres) throws ServletException, IOException{ PrintWriter pw = hres.getWriter(); HttpSession hs = hreq.getSession(true);

```

out.println("<br>Session ID      "+ hs.getId());
out.println("<br>Is New          "+ hs.isNew());
out.println("<br>Creation Time "+ new java.util.Date(hs.getCreationTime()));
}
}

```

c. Write a servlet program **GradeServlet.java** that accepts Grade through radio buttons from **index.html** page, if the string is “A”, “B”, “C” OR “D”, the program should dispatch the direct to the **Success.html** page containing message “Congratulations, You Passed SEM V exam”, else display “Try Again” and load index.html page in current servlet.

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Test extends HttpServlet{
public void doGet(HttpServletRequest hreq, HttpServletResponse hres) throws
ServletException, IOException{
PrintWriter pw = hres.getWriter();
String inp = hreq.getParameter("grade");
if(inp.equals("A") || inp.equals("B") || inp.equals("C") || inp.equals("D") )
{
RequestDispatcher rd = hreq.getRequestDispatcher("Success.html");
rd.forward(hreq, hres);
}
else {
out.println("<H1>TRY Again </h1>");
RequestDispatcher rd = hreq.getRequestDispatcher("index.html");
rd.include(hreq, hres);
}
}
}

```

d. Explain the following w. r. t. working with files in Servlet. **(Reference Book Page No.- 323)**

- i) @MultipartConfigure
- ii) fileSizeThreshold
- iii) location
- iv) maxFileSize
- v) maxRequestSize

1 Marks Each

e. Explain using a code snippet the onDataAvailable() and onAllDataRead() methods of ReadListener interface. **(Reference Book Page No.- 360)**

ReadListener interface description - 1 Marks

onDataAvailable() - **description 1 Mark Code Snippet - 1 Mark**

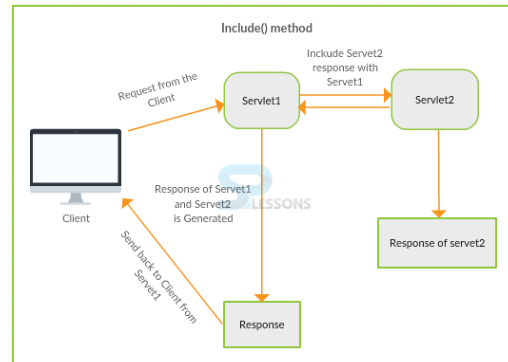
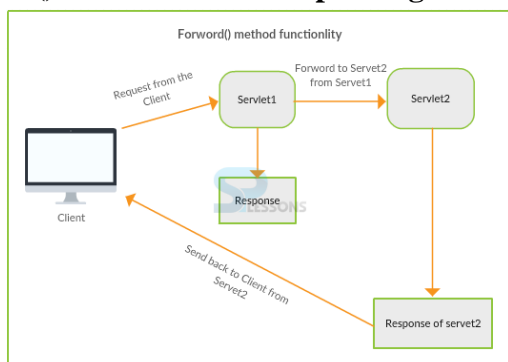
onAllDataRead()- **description 1 Mark Code Snippet - 1 Mark**

f. What is RequestDispatcher? Explain the two methods from RequestDispatcher. **(Reference Book Page No.- 310)**

RequestDispatcher Explanation 1 Marks

forward() method with example/diagram - 2 Marks

include() method with example/diagram - 2 Marks



3.	Attempt <i>any three</i> of the following:
a.	<p>Explain the reason why use JSP? (Reference Book Page No.- 372)</p> <p>Nobody can borrow the code Fast loading of page No Browser compatibility issue Java support Compilation JSP elements in XML or HTML page Any Five - 1 Mark Each</p>
b.	<p>What are directives? Explain page directive with any of its four attributes. (Reference Book Page No.- 390)</p> <p>Def - 1 Marks Explanation of any four attributes of Page Directives - 1 Marks each</p>
c.	<p>Develop a simple JSP application to accept values from html page and display on next page. (Name-txt, age-txt, hobbies-checkbox, email-txt, gender-radio button).</p> <p>HTML Form 2 Marks</p> <pre><HTML> <form action="Show.jsp" >
 Enter Name <input type=text name=txtName >
 Enter Age <input type=text name=txtAge >
 Select Hobbies <input type=checkbox name=txtHob value=Reading >Reading <input type=checkbox name=txtHob value=Singing >Singing
 Select Gender <input type=radio name=txtGender value=Male >Male <input type=radio name=txtGender value=Female >Female <input type=radio name=txtGender value=Other >Other <input type=submit> </form> </HTML></pre> <p>JSP Code -3 Marks</p> <pre>Your Name <%=request.getParameter("txtName") %> Your Age <%=request.getParameter("txtAge") %> <% foreach(i in request.getParameters("txtHob")) out.println("
" +i); %> Gender Selected <%=request.getParameter("txtGender") %></pre>
d.	<p>Explain the <jsp:useBean > tag with its attribute. Support your answer with suitable code snippet. (Reference Book Page No.- 423)</p> <p><usebean> description with attributes - 3 Marks Example - 2 Marks</p>
e.	<p>List the name of JSP implicit objects. Explain any Two in details. (Reference Book Page No.- 429)</p> <p>List - 1 Mark Explanation of any two objects - 4 Marks (2 Marks each)</p>
f.	<p>What is wrong in using JSP scriptlet tag? How JSTL fixes JSP scriptlet shortcomings? (Reference Book Page No.- 474)</p> <p>Problems with JSP tags - 2 Marks How JSTL overcome - 3 Marks</p>
4.	Attempt <i>any three</i> of the following:
a.	<p>Explain benefits of EJB. (Reference Book Page No.- 772)</p> <ul style="list-style-type: none"> • Completely focus of business logic • Reusable components • Portable • Fast building of Application • One Business logic may have multiple presentation logic • Distributed deployment

	<ul style="list-style-type: none"> Application interoperability <p>Any Five - 1 Mark each</p>
b	<p>Write a note on different types of session beans. (Reference Book Page No.- 799)</p> <p>Stateless session bean - 2.5 Marks</p> <p>Stateful session bean - 2.5 Marks</p>
c	<p>Explain life cycle of a message driven bean using suitable diagram. (Reference Book Page No.- 848)</p> <div style="text-align: center;"> <pre> graph LR A[NON EXISTENT] -- "• DI on bean, if any • Create new bean instance • @PostConstruct, if any" --> B[READY] B -- "• @PreDestroy, if any • Destroy the bean instance" --> A B -- "• onMessage" --> B </pre> </div> <p>Diagram - 1 Marks</p> <p>Explanation - 4 Marks</p>
d	<p>Write a stateless session bean code to represent BookInformation. (BookId integer, BookName String, Pages integer, Price double)</p> <pre> package bookdemo; import javax.ejb.Stateless; @Stateless public class BookInfo{ private int bookId, pages; private String bookName; private double price public BookInfo(){ } public void setBookId(int i){bookId=i;} public void setPages(int p){pages=p;} public void setBookName(String n){bookName=n;} public void setPrice(double p){price=p;} public int getBookId(){return bookId;} public int getPages(){return pages;} public String getBookName(){return bookName;} public double getPrice(){return price;} } </pre>
e	<p>What is an interceptor? How an interceptor is defined and how aroundInvoke () is added to it? (Reference Book Page No.- 872)</p> <p>Def and description of interceptor - 2 Mark</p> <p>Working of aroundInvoke() - 3 Marks</p>
f	<p>What is Java Naming and Directory Interface? Explain. (Reference Book Page No.- 892)</p>
<p>5. Attempt any three of the following:</p>	
a.	<p>What is Persistence? (Reference Book Page No.- 902)</p> <p>Persistence is one of the fundamental concepts of application development. It allows DATA to outlive the execution of an application that created it. It is one of the most vital piece of an application without which all the data is simply lost.</p> <p>1 Marks</p> <p>Explanation - 4 Marks</p>
b.	<p>Explain using suitable diagram architecture of Java Persistence API. (Reference Book Page No.- 919)</p>

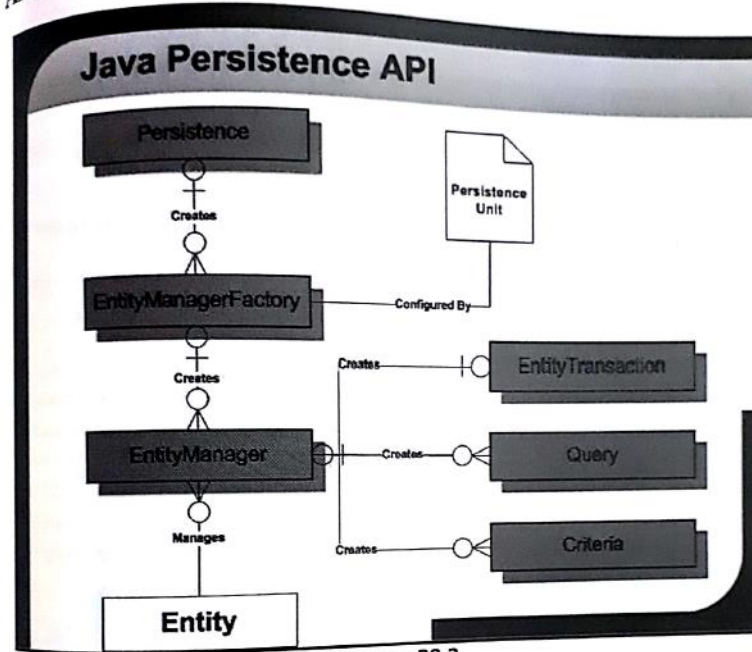


Diagram 38.2

- c. Write a JSP code to add guest feedback using JPA in GuestBook table in database. (Make suitable assumptions)

AddFeedBack.jsp

~~~~~

```
<% @page import="java.util.*,javax.persistence.*,mypack.GuestBook" %>
```

```
<% @page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<%!
```

```
    private EntityManagerFactory entityManagerFactory;
```

```
    private EntityManager entityManager;
```

```
    private EntityManagerTransaction entityManagerTransaction;
```

```
    %>
```

```
<%
```

```
entityManagerFactory = Persistence.createEntityManagerFactory("JPAApplication1PU");
```

```
entityManager = entityManagerFactory.createEntityManager();
```

```
String submit = request.getParameter("btnSubmit");
```

```
    try {
```

```
        String guest = request.getParameter("guest");
```

```
        String message = request.getParameter("message");
```

```
        String messageDate = new java.util.Date().toString();
```

```
        GuestBook gb = new GuestBook();
```

```
        gb.setVisitorName(guest);
```

```
        gb.setMessage(message);
```

```
        gb.setMessageDate(messageDate);
```

```
        entityManagerTransaction = entityManager.getTransaction();
```

```
        entityManagerTransaction.begin();
```

```
        entityManager.persist(gb);
```

```
        entityManagerTransaction.commit();
```

```
    } catch (RuntimeException e) {
```

```
        if(entityManagerTransaction != null) entityManagerTransaction.rollback();
```

```
        throw e;    }
```

```
    try {
```

```
        guestbook = entityManager.createQuery("SELECT g from GuestBook g").getResultList();
```

```
    } catch (RuntimeException e) { }
```

```
entityManager.close();%>
```

d. What is Hibernate? (Reference Book Page No.- 970)

## What Is Hibernate?

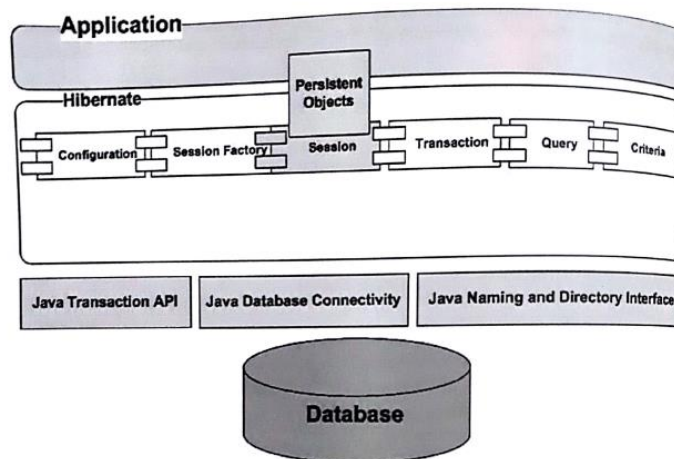
Hibernate is the latest Open Source persistence technology. It is available as a free open source [distributed under the GNU Lesser General Public License] Object/Relational Mapping [ORM] library for the Java programming language. It provides a framework for mapping an object oriented DOMAIN model to a traditional Relational Database.

1 Mark

Explanation 4 Marks

e. Explain using suitable diagram architecture of Hibernate. (Reference Book Page No.- 974)

Architecture Of Hibernate



f. Write a JSP code to add visitor's feedback using Hibernate in FeedBack table in database. (Make suitable assumptions)

```
<%@page import="org.hibernate.*, org.hibernate.cfg.*, mypack.*" %>
<%!
    sessionFactory sf;
    org.hibernate.Session hibSession; %>

<%
sf = new Configuration().configure().buildSessionFactory();
hibSession = sf.openSession();
Transaction tx = null;
GuestBookBean gb = new GuestBookBean();
try{
tx = hibSession.beginTransaction();
String username = request.getParameter("name");
String usermsg = request.getParameter("message");
String nowtime = ""+new java.util.Date();
gb.setVisitorName(username);
gb.setMsg(usermsg);
gb.setMsgDate(nowtime);
hibSession.save(gb);
tx.commit();
out.println("Thank You for your valuable feedback....");
}catch(Exception e){out.println(e);}
hibSession.close();
%>
```