

IAT-2, 7th Sem CSE, JAVA and J2EE 10CS753

Model Answer/Solution

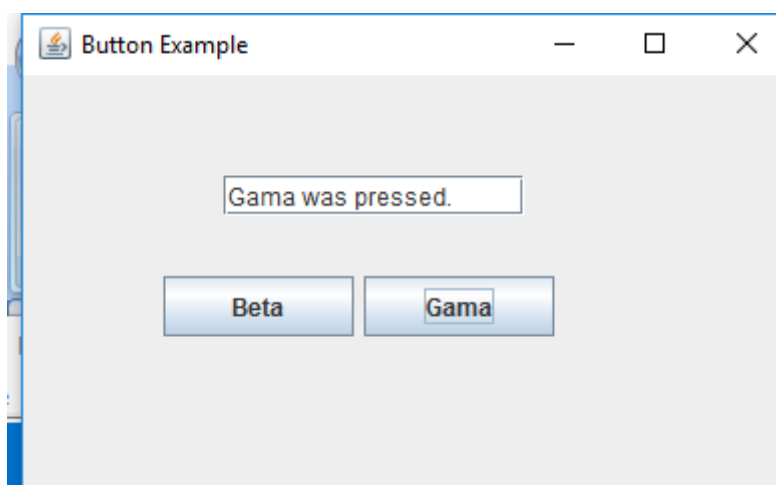
Dr. P. N. Singh, Professor(CSE)

Q1 B Create a Java swing applet that has two buttons name Beta and Gamma. When either of the buttons pressed it should display 'Beta was pressed' and 'Gamma was pressed' respectively. 10 M

Ans:

```
//Beta was Pressed, Gama was pressed
import java.awt.event.*;
import javax.swing.*;
public class BetaGama {
public static void main(String[] args) {
    JFrame f=new JFrame("Button Example");
    final JTextField tf=new JTextField();
    tf.setBounds(100,50, 150,20);
    JButton b=new JButton("Beta");
    b.setBounds(70,100,95,30);
    JButton c=new JButton("Gama");
    c.setBounds(170,100,95,30);
    b.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e){ tf.setText("Beta was pressed."); }
    });
    c.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e){ tf.setText("Gama was pressed."); }
    });
    f.add(tf);
    f.add(b);
    f.add(c);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

Expected Output:



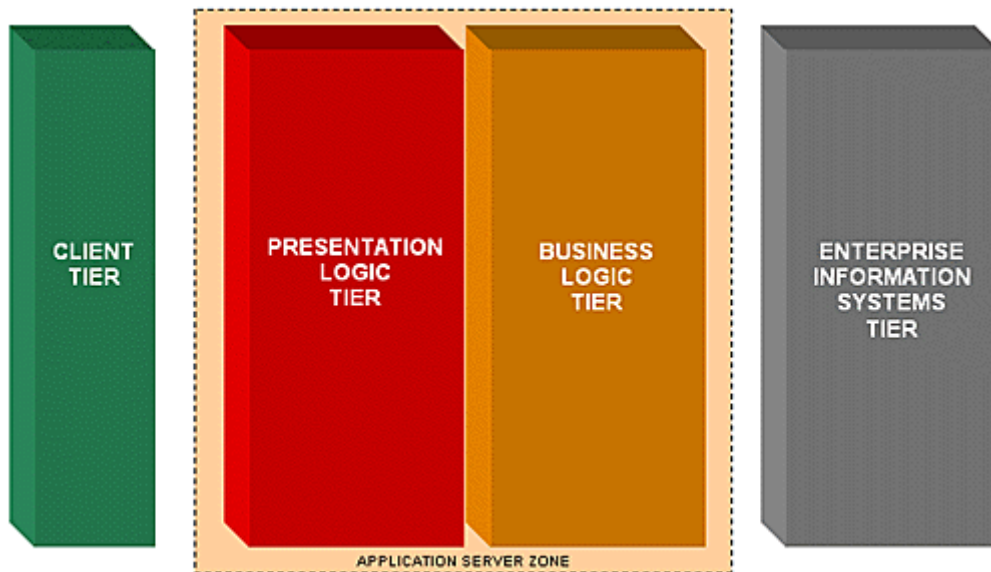
Q 2 (a) Explain the J2EE multitier architecture with a neat diagram.

5M

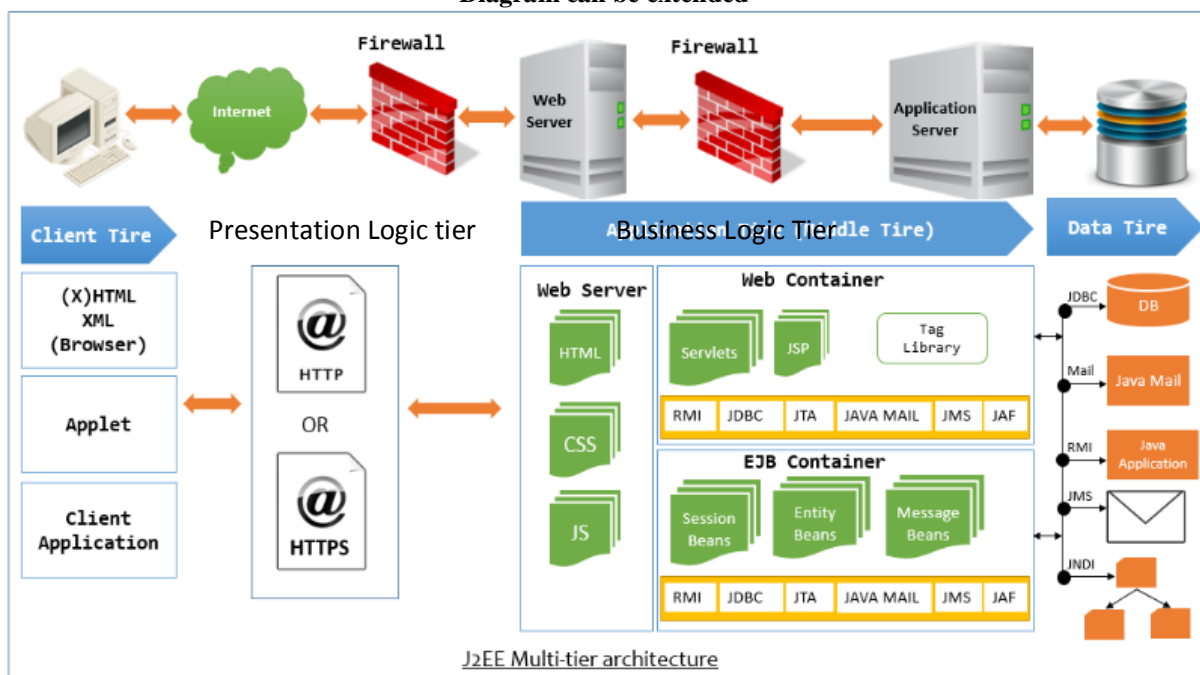
Ans:

J2EE is four-tier architecture. These consist of:

- Client Tier
 - Presentation logic tier,
 - Web tier or Enterprise JavaBeans Tier (or Application server tier or Business logic tier)
 - Enterprise Information Systems Tier or the Data tier.
-
- Client Tier: Prompts & interacts with user. Forward request & receives response
 - Presentation Tier: in http or https format
 - Web/Business logic-Tier: Servlets and JSP (Java Server pages) : accepts request, contains business logic & produced HTML output, Handles concurrency issues for multiple clients
 - Information System/Data Tier: Include variety of resources & connects to resources



J2EE MULTI TIER APPLICATION ARCHITECTURE
Diagram can be extended



J2EE Multi-tier architecture

Q2(b): Write a program to display current contents of a table in the database.

5M

Ans:

```
//Displaying records of Employee table
import java.sql.*;
public class jdbcResultSet {
    public static void main(String[] args) {
        try {
            Class.forName("org.apache.derby.jdbc.ClientDriver");
        } catch(ClassNotFoundException e) {
            System.out.println("Class not found "+ e);
        }
        try {
            Connection con = DriverManager.getConnection(
                "jdbc:derby://localhost:1527/testDb","username", "password");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM employee");
            System.out.println("id name job");
            while (rs.next()) {
                int id = rs.getInt("id");
                String name = rs.getString("name");
                String job = rs.getString("job");
                System.out.println(id+" "+name+" "+job);
            }
        } catch(SQLException e) {
            System.out.println("SQL exception occured" + e);
        }
    }
}
```

Expected Output

ID	Name	Job
P001	Dr. Kumar Swami	Professor
P002	Sadhana Satpati	Programmer
P003	Shreedhar K.	Asst. Professor

.....

Q3 What are transactions & transaction properties? Write a Java program to demonstrate how to process a database transaction.

10 M

Ans:

Transaction represents a **single unit of work**.

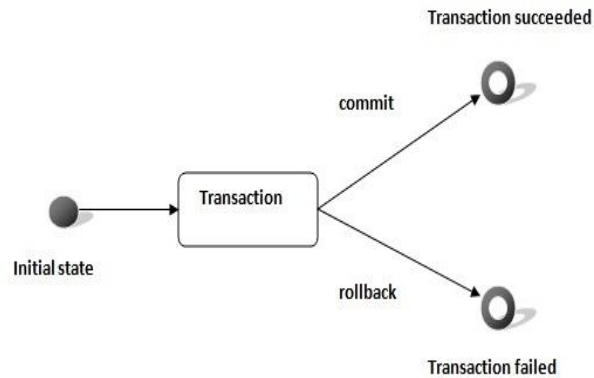
The ACID properties describes the transaction management well. ACID stands for Atomicity, Consistency, isolation and durability.

Atomicity means either all successful or none.

Consistency ensures bringing the database from one consistent state to another consistent state.

Isolation ensures that transaction is isolated from other transaction.

Durability means once a transaction has been committed, it will remain so, even in the event of errors, power loss etc.



```

// Program to update/add records
import java.sql.*;
class FetchRecords{
  public static void main(String args[])throws Exception{
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
    con.setAutoCommit(false);
    Statement stmt=con.createStatement();
    stmt.executeUpdate("insert into employee values(201,'Chirayu',40000)");
    stmt.executeUpdate("insert into employee values(202,'Navya',50000)");
    con.commit();
    con.close();
  }
}

```

Q4 What are servlets & its application in web programming? Explain the different stages in life cycle of a servlet API's. 10 M

Ans:

Servlet technology is used to create web application (resides at server side and generates dynamic web page). **Servlet** technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was popular as a server-side programming language.

Servlet can be described in many ways, depending on the context.

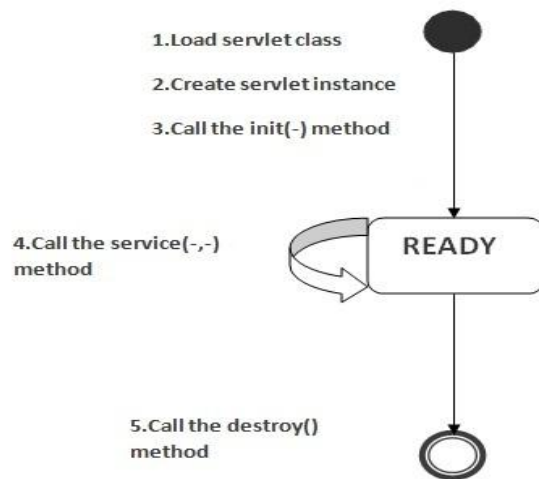
- Servlet is a technology i.e. used to create web application.
- Servlet is an API that provides many interfaces and classes including documentations.
- Servlet is an interface that must be implemented for creating any servlet.
- Servlet is a class that extend the capabilities of the servers and respond to the incoming request. It can respond to any type of requests.
- Servlet is a web component that is deployed on the server to create dynamic web page.

The web container maintains the life cycle of a servlet instance.

- Servlet class is loaded.
- Servlet instance is created.
- init method is invoked.
- service method is invoked.
- destroy method is invoked.

3 states of a servlet: new, ready and end.

1. The servlet is in new state if servlet instance is created.
2. After invoking the init() method, Servlet comes in the ready state. In the ready state, servlet performs all the tasks.
3. When the web container invokes the destroy() method, it shifts to the end state.



Q5 List out & explain methods defined by Servlet cookie and various cookies attributes 10 M

Ans:

To create a cookie:

```
Cookie ck=new Cookie("user","Vikram Singh");//creating cookie object
response.addCookie(ck);//adding cookie in the response
```

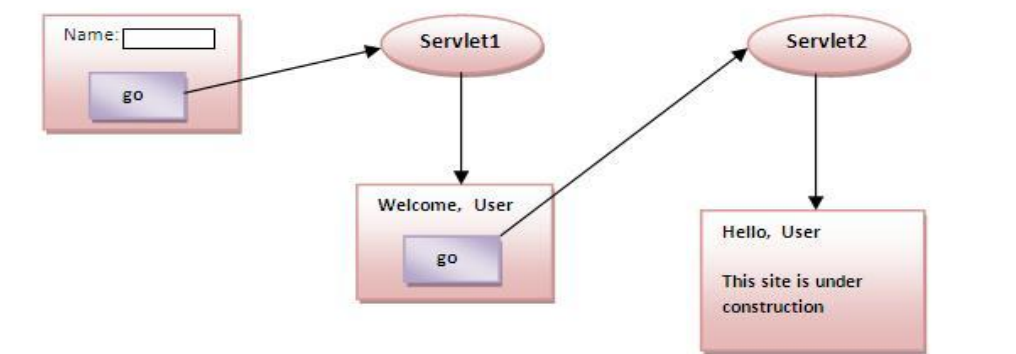
To delete a cookie:

```
Cookie ck=new Cookie("user","");//deleting value of cookie
ck.setMaxAge(0);//changing the maximum age to 0 seconds
response.addCookie(ck);//adding cookie in the response
```

To get all cookies:

```
Cookie ck[]=request.getCookies();
for(int i=0;i<ck.length;i++){
    out.print("<br>" +ck[i].getName()+" "+ck[i].getValue());
    //printing name and value of cookie
}
```

A **cookie** has a name, a single value, and optional **attributes** such as a comment, path and domain qualifiers, a maximum age, and a version number. Some Web browsers have bugs in how they handle the optional **attributes**, so those should be used carefully to improve the interoperability of the **servlets**.



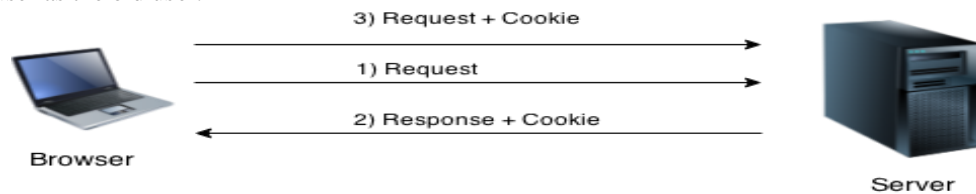
Q6. Explain the uses of cookies with servlet programs. Write a servlet program to add a cookie. 10 M

Ans:

- Cookies are small bits of textual information that a Web server sends to a browser and that the browser returns unchanged when visiting the same Web site or domain later.
- A cookie is a small piece of information that is persisted between the multiple client requests.
- A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

How cookies work:

- By default, each request is considered as a new request. In cookies technique, we add cookie with response from the servlet.
- So cookie is stored in the cache of the browser.
- After that if request is sent by the user, cookie is added with request by default. Thus, we recognize the user as the old user.



//Example servlet program and to add cookie

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class FirstServlet extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response){
        try{
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            String n=request.getParameter("userName");
            out.print("Welcome "+n);
            Cookie ck=new Cookie("uname",n);//creating cookie object
            response.addCookie(ck);//adding cookie in the response
            //creating submit button
            out.print("<form action='servlet2'>");
            out.print("<input type='submit' value='go'>");
            out.print("</form>");
            out.close();
        }catch(Exception e){System.out.println(e);}
    }
}
  
```

Q7 Briefly explain prepared statement & callable statement object. Write a program to call stored procedure using callable statement. 10M

Ans:

There are 3 types of Statements, as given below:

Statement:

It can be used for general-purpose access to the database. It is useful when we are using static SQL statements at runtime.

PreparedStatement:

It can be used when we plan to use the same SQL statement many times. The PreparedStatement interface accepts input parameters at runtime.

CallableStatement:

CallableStatement can be used when we want to access database stored procedures.

- CallableStatement interface is used to call the **stored procedures and functions**.

- We can have business logic on the database by the use of stored procedures and functions that will make the performance better because these are precompiled.
- may create a function that receives date as the input and returns age of the employee as the output.
- The prepareCall() method of Connection interface returns the instance of CallableStatement.
- public CallableStatement prepareCall("{ call procedurename(?,?...?)}");

CallableStatement stmt=con.prepareCall("{ call myprocedure(?,?)}");

Assuming that stored procedure is:

```
create or replace procedure "INSERT"
(id IN NUMBER,
name IN VARCHAR2)
is
begin
insert into user420 values(id,name);
end;
```

And table structure is:

```
create table user420(id number(10), name varchar2(200))
```

```
import java.sql.CallableStatement;
```

...

```
//
// Prepare a call to the stored procedure 'demoSp'
// with two parameters
// Notice the use of JDBC-escape syntax ({call ...})
```

```
CallableStatement cStmt = conn.prepareCall("{call demoSp(?, ?)}");
```

```
cStmt.setString(1, "abcdefg");
```

Q8 (a): What is meant by scrollable result set?

5 M

Ans:

- Whenever we create an object of ResultSet by default, it allows us to retrieve in forward direction only and we cannot perform any modifications on ResultSet object.
- Therefore, by default the ResultSet object is non-scrollable and non-updatable ResultSet.
- But we can make this object to move forward and backward direction by passing either TYPE_SCROLL_INSENSITIVE or TYPE_SCROLL_SENSITIVE in createStatement(int,int) method as well as we can make this object as updatable by:

```
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_UPDATABLE);
```

The value of Type and value of Mode are present in ResultSet interface as constant data members and they are:

int Type	int Mode
TYPE_FORWARD_ONLY - 1	CONCUR_READ_ONLY - 3
TYPE_SCROLL_INSENSITIVE - 2	

Q8(b): Analyze differences between Servlets & JSP.

5M

Ans:

Difference between servlets & JSP

- I. JSP is for web page and servlet is for java components.

- II. A servlet is HTML in java where as JSP is java in HTML
- III. A servlet is like any other java class. We put HTML into print statement like we use System.out or how java scripts uses document.write
- IV. A JSP technology gets converted to a servlet but it looks more like php.
- V. Servlets run faster compared to JSP. JSP can be compiled into java servlets.
- VI. JSP is web page scripting language that can generate dynamic content while servlets are java programs that are already compiled which creates dynamic web content