Biyani's Think Tank

Concept based notes

Java Technologies

MCA

Ms Ritu

Revised by: Mr Shiv Kishore Sharma

Deptt. of IT

Biyani Girls College, Jaipur

Published by :
Think Tanks
Biyani Group of Colleges

Edition : 2012

Leaser Type Setted by :
Biyani College Printing Department

Preface

I am glad to present this book, especially designed to serve the needs of the students. The book has been written keeping in mind the general weakness in understanding the fundamental concepts of the topics. The book is self-explanatory and adopts the “Teach Yourself” style. It is based on question-answer pattern. The language of book is quite easy and understandable based on scientific approach.

Any further improvement in the contents of the book by making corrections, omission and inclusion is keen to be achieved based on suggestions from the readers for which the author shall be obliged.

I acknowledge special thanks to Mr. Rajeev Biyani, *Chairman* & Dr. Sanjay Biyani, *Director* (*Acad.*) Biyani Group of Colleges, who are the backbones and main concept provider and also have been constant source of motivation throughout this endeavour. They played an active role in coordinating the various stages of this endeavour and spearheaded the publishing work.

I look forward to receiving valuable suggestions from professors of various educational institutions, other faculty members and students for improvement of the quality of the book. The reader may feel free to send in their comments and suggestions to the under mentioned address.

Author

Syllabus

Introduction to Java Enterprise, API JDBC, fundamentals, J2EE multi-tier architecture, Web Applications in J2EE.

Servlets fundamentals – architecture, life cycle of a servlet, initialization, threads, servlets and HTML, retrieving data in servlet, servicing he GET and POST requests,

servlet sessions – session tracking, cookies.

Servlets, JDBC and Inter servlet communications – JDBC, Driver types, JDBC servlet, JDBC connection pool, inter servlet communication, servlet security and different packages of JSP and servlets.

JSP fundamentals – architecture, implicit objects, standard actions, JSP errors.
J2ME – introduction, building MIDlets, creating a user interface, event handling with commands, tickers, screens, textbox, lists and forms.

S. No.	Name of Topic
1.	Unit I: **Java 2 Enterprise Edition**
	1.1 Introduction
	1.2 Components
	1.3 Multi-tier Architecture
2.	Unit II **: Servlet**
	2.1 Introduction and life cycle of servlet
	2.2 Single-Thread model
	2.3 Difference between GET and POST method
	2.4 Example of servlet
	2.5 Session Tracking
	2.6 Example of Session
	2.7 Cookies with Example
	2.8 Inter-servlet Communication
3.	Unit III: **Java Database Connectivity**
	3.1 Introduction
	3.2 Steps involved in basic jdbc operation
	3.3 Types of jdbc drivers
	3.4 Connection Pooling
	3.5 Servlet Security
4.	Unit IV: **Java Server Pages(JSP)**
	4.1 Introduction and Architecture
	4.2 Implicit objects
	4.3 Standard Actions
	4.4 JSP Errors

Unit 1
J2EE

MCQ's

1. Your application supports multiple client types including HTTP clients. Your business layer is implemented using Enterprise Java Beans. Which of the following is best suited for maintaining client state?
   a. Stateful session beans
   b. Entity Beans
   c. HttpSession attributes
   d. Cookies
   e. URL Rewriting

Correct answer: a

2. Which enterprise bean type is defined without any client view interfaces?
   a. BMP Entity bean
   b. CMP Entity bean
   c. Stateful Session bean
   d. Stateless Session bean
   e. Message Driven Bean

Correct answer: e

3. Which of the following should NOT be used to share data between servlets in a distributed web application?
   a. Attributes of ServletContext
   b. Enterprise Java Beans
   c. Attributes of HttpSession
   d. Database

Correct answer: a

4. You want to use URL rewriting to support client browsers, which do not use cookies. Which method will you use to attach the session id to a URL that is to be used for the sendRedirect() method of the HttpServletResponse?
   a. encodeURL
   b. encodeRedirectURL
   c. encodeSessionURL
   d. encodeSessionRedirectURL
   e. None of these

Correct answer: b

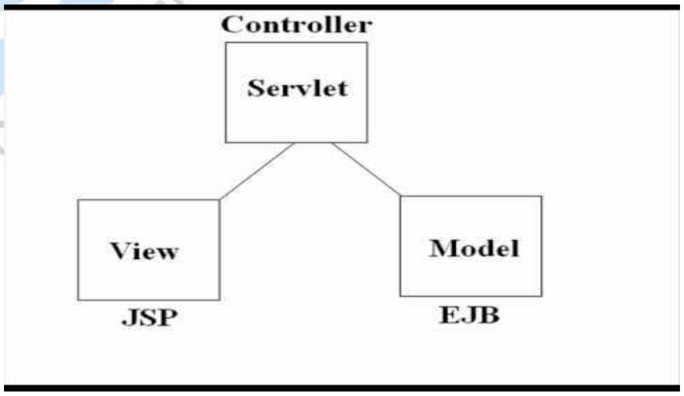
7. Which of these packaging options of enterprise beans are recommended for most J2EE applications?
   a. Package each enterprise bean for an application in its own EJB module
   b. Package all enterprise beans for an application in one EJB module
   c. Package all related enterprise beans for an application in one EJB module
   d. None of These

Correct answer: c

Q:1 What is Java Enterprise ?

Ans: Java Enterprise uses a component based approach for the enterprise application. It includes:

- Design
- Development
- Assembly
- Deployment of enterprise application.

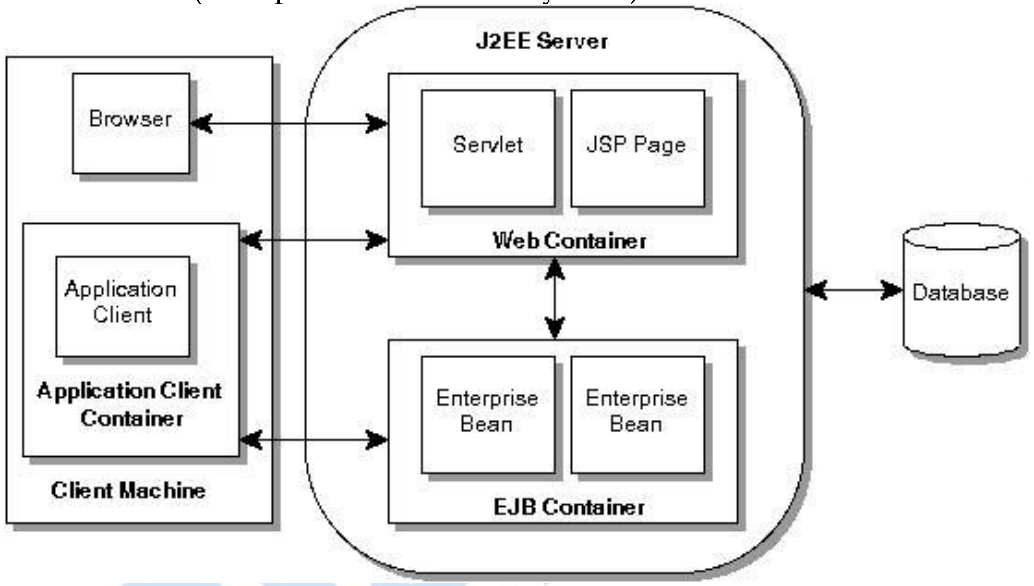


Java Enterprise

Q:2 What are the components of Java Enterprise ?

Ans There are 4 tiers in java enterprise:

1. Client-tier component: Run on the client machine
2. Web-tier component: Run on the java EE server
3. Business-tier component: Run on the java EE server
4. EIS-tier(enterprise information syatem): Run on the EIS server



JAVA ENTERPRISE

Q:3 Explain the multi-tier architecture of J2EE?

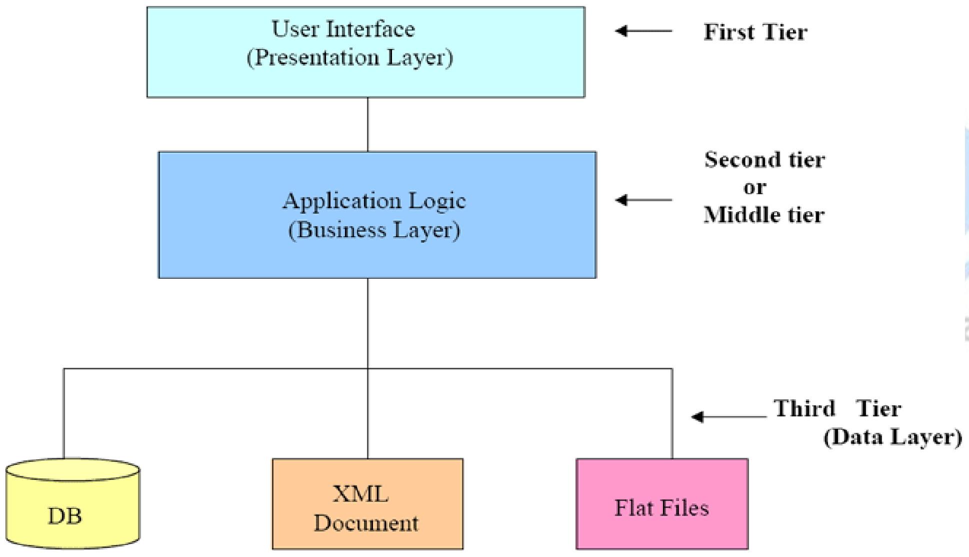
Ans J2EE architecture uses the component-based development of multi-tier enterprise applications.

A J2EE application system includes:

- **Client tier**: In this tier Servlets and JavaServer Pages (JSPs), or standalone Java applications provide a dynamic interface to the middle tier.

- **Middle tier**:, In this tier enterprise beans and Web Services encapsulate reusable, distributable business logic for the application.

- **Enterprise data tier**: In the data tier, the enterprise's data is stored and persisted, typically in a relational database.



Typical Three Tier Architecture

Unit 2

Servlets

MCQ's

1. If the HTTP error 500 is generated by your servlet, you do not want to show the "Internal Server Error" page to the client. Instead, you want a custom error page to be displayed. What is the best way to accomplish this?
   a. Forward the user to the error page using HttpServletResponse.sendRedirect method.
   b. Forward the user to the error page using RequestDispatcher.forward method.
   c. Specify the mapping of the error-code 500 and the error page in the deployment descriptor.
   d. It is not possible to accomplish this.

Correct answer: c

2. A servlet needs to acquire a data source through a JNDI naming lookup. Which of the following is the best place to do this?
   a. Constructor
   b. init method
   c. service method
   d. doGet method
   e. doPost metho

Correct answer: b

3. What's the difference between servlets and applets?
   a. Servlets executes on Servers, where as Applets executes on Browser
   b. Servlets have no GUI, where as an Applet has GUI
   c. Servlets creates static web pages, where as Applets creates dynamic web pages

d. Servlets can handle only a single request, where as Applet can handle multiple requests

Correct answer: a and b

4. Which method cannot be called by a stateful session bean using bean-managed transactions?
   a. getUserTransaction()
   b. getRollbackOnly()
   c. getStatus()
   d. None of these

Correct answer: b

5. Which method is used to specify before any lines that uses the PintWriter?
   a. setPageType()
   b. setContextType()
   c. setContentType()
   d. setResponseType()

Correct answer: c

6. Which of the following are the session tracking techniques?
   a. URL rewriting, using session object, using response object, using hidden fields
   b. URL rewriting, using session object, using cookies, using hidden fields
   c. URL rewriting, using servlet object, using response object, using cookies
   d. URL rewriting, using request object, using response object, using session object

Correct answer: b

7. The getSession() method with 'true' as its parameter [getSession(true)] it will return the appropriate session object when
   a. the session is completed
   b. the session object is passed to another method
   c. the session does not exists
   d. the session is existing

Correct answer: d

8. The tasks – authentication-blocking of requests, data compression, logging and auditing – are performed by
   a. servlet filter
   b. servlet config
   c. servlet context
   d. servlet container

Correct answer: a

9. The method forward(request,response) will
   a. return back to the same method from where the forward was invoked
   b. not return back to the same method from where the forward was invoked and the web pages navigation continues
   c. Both A and B are correct
   d. None of the above

Correct answer: a

10. A servlet maintain session in
   a. Servlet container
   b. Servlet context
   c. Servlet request heap
   d. Servlet response heap

Correct answer: b

11. Servlet mapping defines
   a. an association between a URL pattern and a servlet
   b. an association between a URL pattern and a request page
   c. an association between a URL pattern and a response page
   d. All of the above

Correct answer: a

12. The life cycle of a servlet is managed by
   a. servlet context
   b. servlet container
   c. the supporting protocol (such as http or https)
   d .all of the above

Correct answer: b

13. The init parameter name and value pairs that are defined in web.xml file are handled by
    a. ServletConfig object
    b. ServletContext object
    c. ServletRequest object
    d. ServletResponse object

Correct answer: a

14. How many ServletContext objects are available for an entire web application?
    a. One each per servlet
    b. One each per request
    c. One each per response
    d. Only one

Correct answer: d

15. What are the mechanisms available in ServletContextListener interface?
    a. contextInit(), contextService(), contextDestroyed()
    b. contextInitialized((),contextDestroyed()
    c. contextInitialized(), contextService(), contextDestroyed()
    d. None of the above

Correct answer: b

Q:1 Explain servlet and life cycle of servlet?

Ans

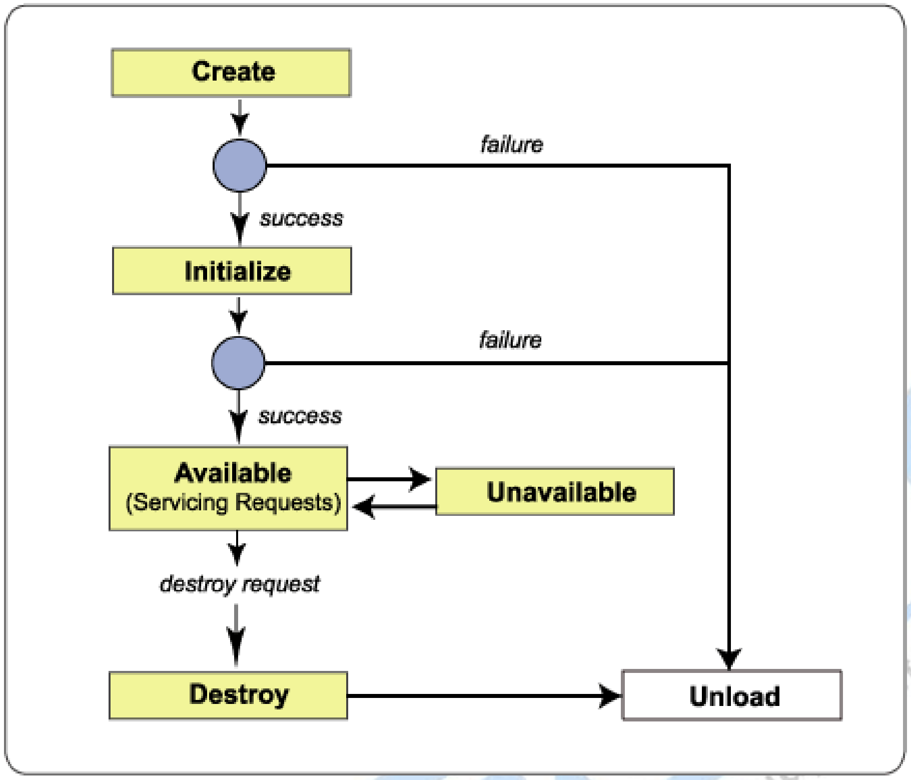
Servlet:

- servlets are small java programs that are used to extend the functionality of the server.
- Always save and run on the server

Life cycle of the servlet:

The life cycle of a servlet can be categorized into four parts:

- Loading and Instantiation
- Initialization
- Servicing the Request
- Destroying the Servlet



Life cycle of the servlet

1. **Loading and Instantiation:** The servlet is load with loading of the container otherwise it load when the first request comes for service. After loading of the servlet, the container creates the instances of the servlet.

2. **Initialization:** After creating the instances, the servlet container calls the init( ) method and passes the servlet initialization parameters to the init( ) method.

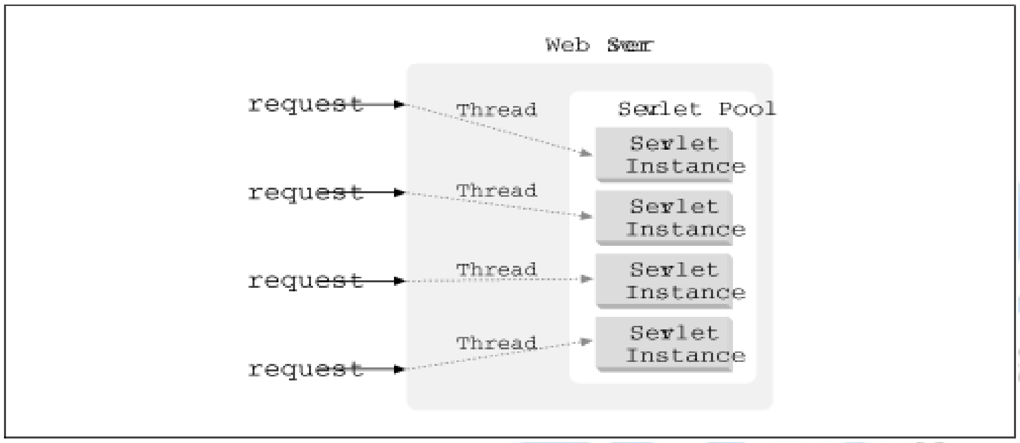
3. **Servicing the Request:** The sevlet container calls the service() method for servicing any request. The service() method determines the kind of request and calls the appropriate method (doGet() or doPost()) for handling the request and sends response to the client using the methods of the response object.

4. **Destroying the Servlet:** If the servlet is no longer needed for servicing any request, the servlet container calls the destroy() method .

Q: 2 What is Single –Thread model in servlet?

Ans

- Servlet creates single thread for each request of client.
- Each thread has a single flow of execution.



Single –Thread model

- A server that loads a SingleThreadModel servlet must guarantee documentation that no two threads will execute concurrently the service method of that servlet.

- Each thread uses a free servlet instance from the poo.

- Thus, any servlet implementing SingleThreadModel can be considered thread safe and isn't required to synchronize access to its instance variables.

Q: 3 What is difference between GET and POST method?

Ans

In GET method
- data is submitted as a part of url
- data is visible to the user
- it is not secure but fast and quick

In POST method
- data is submitted as a part of http request
- data is not visible in the url
- it is more secure but slower as compared to GET

Q:4 Give one example of Servlet?

Ans

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld extends HttpServlet
 {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException
  {
  response.setContentType("text/html");
  PrintWriter out = response.getWriter();
  out.println("<html>");
  out.println("<head>");
  out.println("<title>Hello World!</title>");
  out.println("</head>");
  out.println("<body>");
  out.println("<h1>Hello World!</h1>");
  out.println("</body>");
 out.println("</html>");
   }
}
```

Output:

Hello World!

Q:5 What is Session Tracking?

Ans

- When multiple clients make continuous request the server cannot identify from which client it is getting requests.For this purpose session is used.
- Because HTTP is a stateless protocol.When there is a need to maintain the conversational state, session tracking is needed.

Session tracking methods:

1. User authorization
2. Hidden fields

3. URL rewriting
4. Cookies
5. Session tracking API

1. User Authorization

The user has username and password to login to the application. Based on that the user can be identified and the session can be maintained.

2. Hidden Fields

<INPUT TYPE="hidden" NAME="technology" VALUE="servlet">
Hidden fields like the above can be inserted in the webpages and information can be sent to the server for session tracking.

3. URL Rewriting

Original URL: http://server:port/servlet/ServletName
Rewritten URL: http://server:port/servlet/ServletName?sessionid=7456
When a request is made, additional parameter is appended with the url.

4. Cookies

Cookie is a key value pair of information, sent by the server to the browser.

Q:5 Explain Session with example?

Ans

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class SessionExample extends HttpServlet
{
public void doGet(HttpServletRequest request, HttpServletResponse response)
 throws IOException, ServletException
 {
 response.setContentType("text/html");
  PrintWriter out = response.getWriter();

 HttpSession session = request.getSession(true);
```

```
      // print session info

     Date created = new Date(session.getCreationTime());
   Date accessed = new Date(session.getLastAccessedTime());
     out.println("ID " + session.getId());
    out.println("Created: " + created);
     out.println("Last Accessed: " + accessed);

     // set session info if needed

    String dataName = request.getParameter("dataName");
   if (dataName != null && dataName.length() > 0) {
        String dataValue = request.getParameter("dataValue");
   session.setAttribute(dataName, dataValue);
     }

     // print session contents

    Enumeration e = session.getAttributeNames();
    while (e.hasMoreElements()) {
       String name = (String)e.nextElement();
       String value = session.getAttribute(name).toString();
       out.println(name + " = " + value);
       }
    }
   }
```

Output:

```
Session ID: D91A214517E7F08C01A755737D170F41
Created: Wed Jul 04 10:07:34 CEST 2012
Last Accessed: Wed Jul 04 10:19:54 CEST 2012

The following data is in your session:
```

Name of Session Attribute:

Value of Session Attribute:

GET based form:

Name of Session Attribute:

Value of Session Attribute:

Q6 What are cookies? Explain with example?

Ans

- Cookie is a text file.
- It is saved in key value pair of information.
- It is sent by the server to the browser.
- This should be saved by the browser in its space in the client computer.
- Whenever the browser sends a request to that server it sends the cookie along with it. Then the server can identify the client using the cookie.

Example:

```
Cookie cookie = new Cookie(“userID”, “7456");
res.addCookie(cookie);
```

Example of Cookie:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class UseCookies extends HttpServlet

{

public void doGet ( HttpServletRequest request,HttpServletResponse response )
throws ServletException, IOException {
  PrintWriter out;
  response.setContentType("text/html");
  out = response.getWriter();
```

```
         Cookie cookie = new Cookie("CName","Cookie Value");
        cookie.setMaxAge(100);
         response.addCookie(cookie);

        out.println("<HTML><HEAD><TITLE>");
         out.println(" Use of cookie in servlet");
        out.println("</TITLE></HEAD><BODY BGCOLOR='cyan'>");
        out.println(" <b>This is a Cookie example</b>");
         out.println("</BODY></HTML>");
         out.close();
         }
}
```

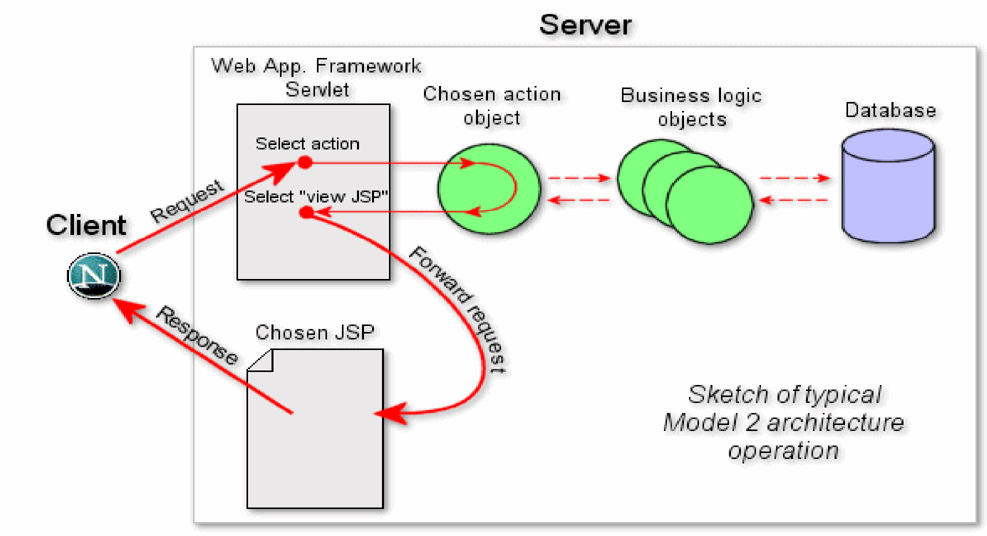
Output:

```
This is a Cookie example
```

Q:7 What is inter servlet communication?

Ans

Inter servlet communication means communication between servlets. **ways to communicate between servlets:**



- Request Dispatching
- HTTP Redirect
- Servlet Chaining
- HTTP request (using sockets or the URLConnection class)
- Shared session, request, or application objects (beans)
- Direct method invocation (deprecated)
- Shared static or instance variables (deprecated)

Q:8 What is servlet security?

Ans

Servlet security includes:

1. HTTP Authentication
2. Digital Certificates
3. Secure Sockets Layer (SSL)
4. Running Servlets Securely

1.HTTP Authentication:

The HTTP protocol provides built-in authentication support--called basic authentication--based on a simple challenge/response, username/password model

2.Digital Certificates:

A simple servlet that tells the client its name and what kind of authentication has been performed (basic, digest, or some alternative).
Methods:

- getRemoteUser()
- getAuthType()

3.Secure Sockets Layer (SSL):

The server administrator tells the server which resources are to be restricted to which users, and information about those users (such as their passwords) is somehow made available to the server.

Unit 3
JDBC

MCQ's

1. The JDBC-ODBC Bridge supports multiple concurrent open statements per connection?
   a. True
   b. False

Correct answer: a

2. Which driver is efficient and always preferable for using JDBC applications?
   a. Type – 4
   b. Type – 1
   c. Type – 3
   d. Type – 2

Correct answer: a

3. JDBC facilitates to store the java objects by using which of the methods of PreparedStatement
1. setObject () 2. setBlob() 3. setClob()
   a. 1, 2
   b. 1,2,3
   c. 1,3
   d. 2,3

Correct answer: b

4. The JDBC-ODBC bridge is
a. Three tiered
b. Multithreaded
c. Best for any platform
d. All of the above

Correct answer: b

5. All raw data types (including binary documents or images) should be read and uploaded to the database as an array of
   a. byte
   b. int
   c. boolean
   d. char

Correct answer: a

6. The class java.sql.Timestamp has its super class as
   a. java.sql.Time
   b. java.util.Date
   c. java.util.Time
   d. None of the above

Correct answer: b

7. Which of the following methods finds the maximum number of connections that a specific driver can obtain?
   a. Database.getMaxConnections
   b. Connection.getMaxConnections
   c. DatabaseMetaData.getMaxConnections
   d. ResultSetMetaData.getMaxConnections

Correct answer: c

8. Are prepared statements actually compiled?
   a. Yes, they compiled
   b. No, they are bound by the JDBC driver

Correct answer: a

9. When the message "No Suitable Driver" occurs?
   a. When the driver is not registered by Class.forname() method
   b. When the user name, password and the database does not match
   c. When the JDBC database URL passed is not constructed properly
   d. When the type 4 driver is used

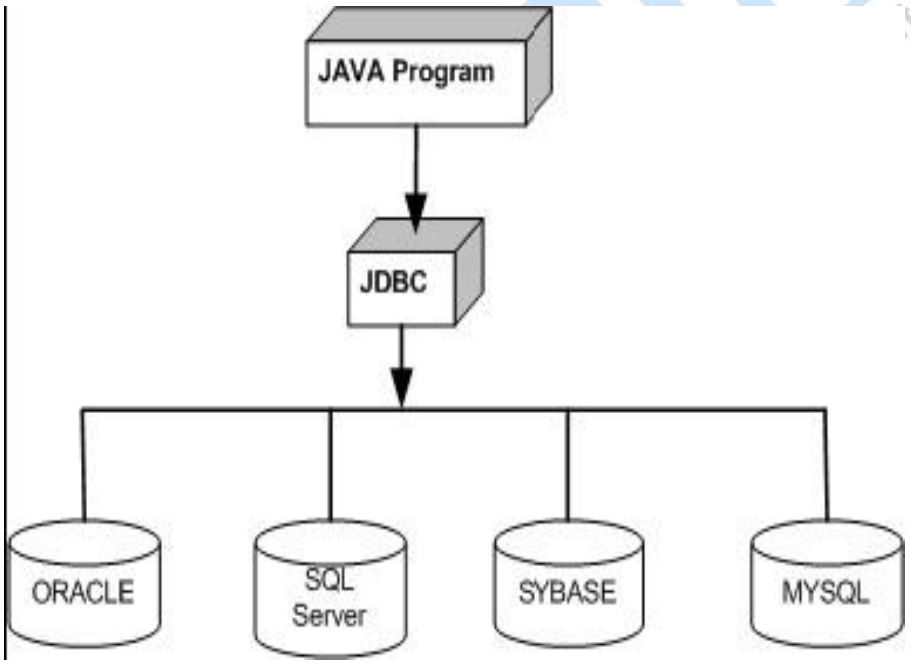
Correct answer: c

10. How many transaction isolation levels are defined in java.sql.Connection interface?
    a. 4
    b. 3
    c. 5
    d. 2

Correct answer: c

Q:1 What is Java Database Connectivity (JDBC)?
Ans

- Java Database Connectivity (JDBC) is an application program interface (API).
- connect programs written in Java to the data in popular database.
- JDBC allows a developer to write applications that is database independent.
- JDBC allows you to write Java code, and leave the platform (database) specific code to the driver.



JDBC

JDBC concepts:

- **Statement**: An SQL Statement to perform a query or update operation
- **Metadata**: Information about returned data, the database and the driver

- **ResultSet**: Logical set of columns and rows returned by executing an SQLstatement .

Q:2 What are the steps involved in basic JDBC operations?
Ans

Steps Involved in Basic JDBC Operations:

1. Load the JDBC driver class:
```
Class.forName("driverName");
```

2. Open a database connection:
```
DriverManager.getConnection
("jdbc:xxx:datasource");
```

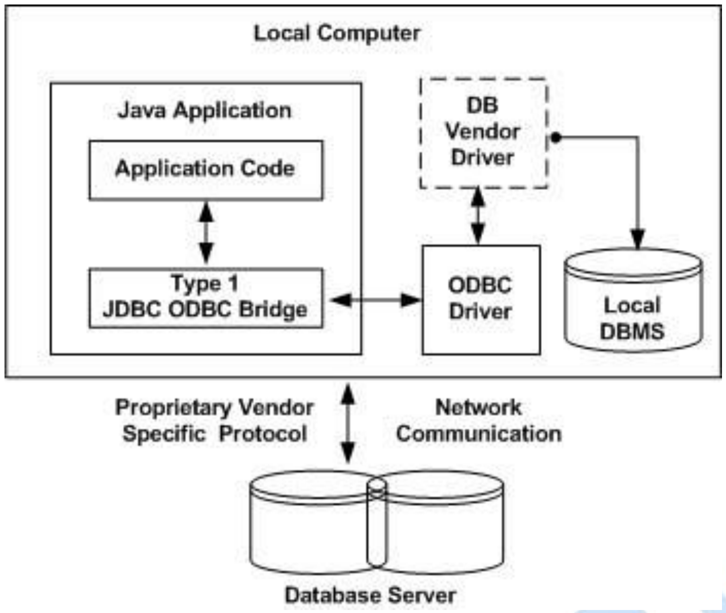
3. Issue SQL statements:
```
stmt = con.createStatement();
stmt.executeQuery ("Select * from myTable");
```

4. Process result set:
```
while (rs.next()) {
name = rs.getString("name");
amount = rs.getInt("amt"); }
```

Q:3 What are the types of JDBC Drivers?
Ans

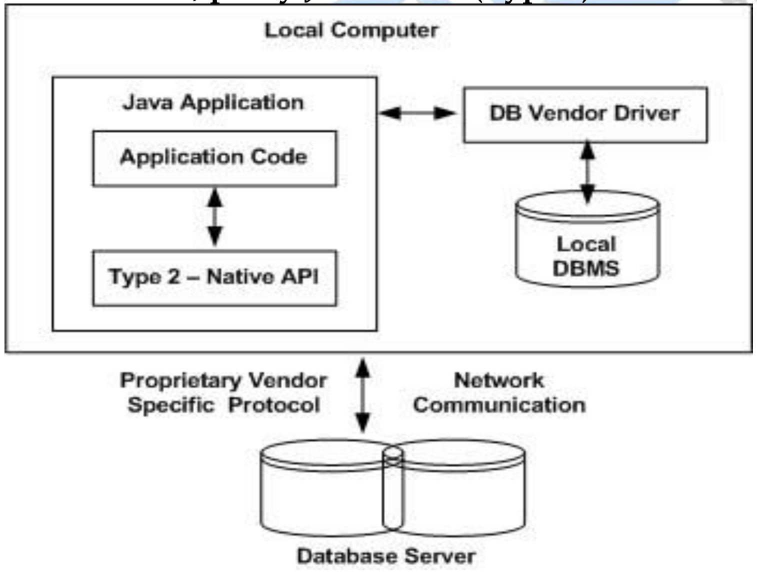
- JDBC-ODBC Bridge, plus ODBC driver (Type 1)
- Native-API, partly Java driver (Type 2)
- JDBC-net, pure Java driver (Type 3)
- Native-protocol, pure Java driver (Type 4)

1. JDBC-ODBC Bridge, plus ODBC driver (Type 1)



- Provides JDBC access to databases through ODBC drivers
- ODBC driver must be configured for the bridge to work
- Only solution if no JDBC driver available for the DBMS

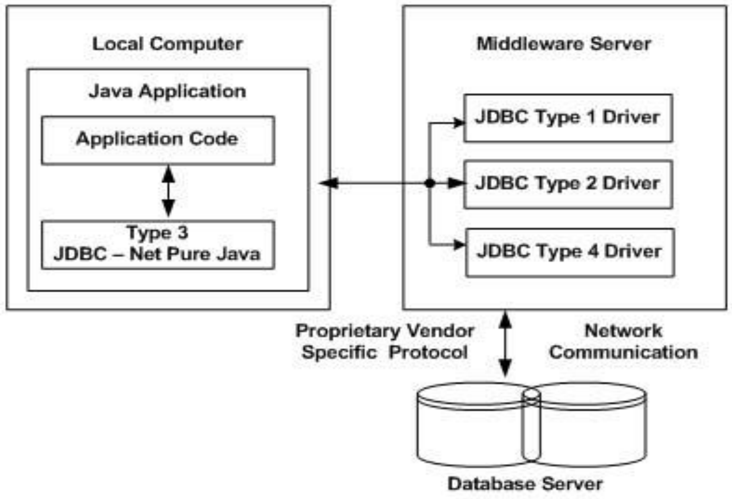
2.Native-API, partly Java driver (Type 2)



- Native-API driver converts JDBC commands into DBMS-specific native calls
- Same restrictions as Type1 – must have some binary code loaded on its machine

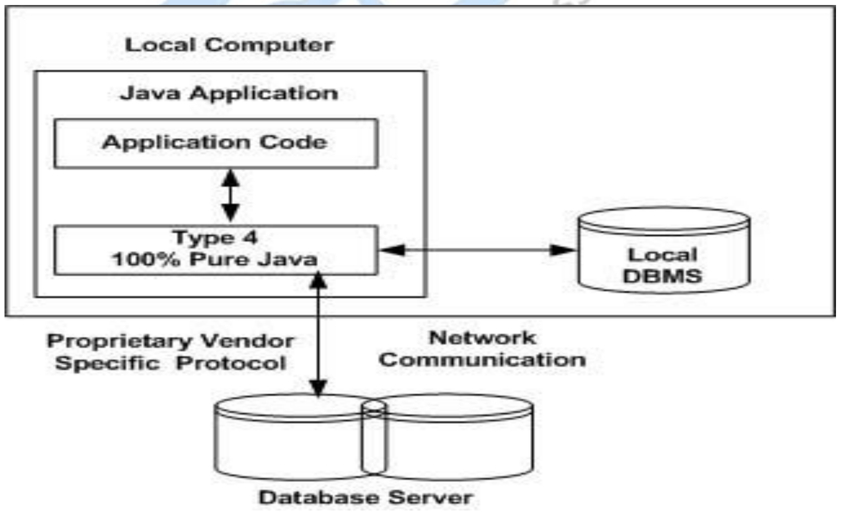
- Directly interfaces with the database

3.JDBC-net, pure Java driver (Type 3)



- Translates JDBC calls into a database-independent network protocol and sent to a middleware server.
- This server translates this DBMS-independent protocol into a DBMS-specific protocol and sent to the database
- Results sent back to the middleware and routed to the client

4. Native-protocol, pure Java driver (Type 4)



- Pure Java drivers that communicate directly with the vendor's database
- JDBC commands converted to database engine's native protocol directly
- Advantage: no additional translation or middleware layer
- Improves performance

Q:4 What is Connection pooling and why we use this?
Ans

- Database connection is very expensive to open and close.
- The valuable database resources such as memory, cursors, locks , temporary tables all tends to increase on numbers of concurrent connections.
- In connection pooling, we creates limited numbers of connection objects pools at a time, such as 10 connections, 50 connections, 100 connections etc. This depends upon the capacity of the database that how much connections it can handle at a time. If any request comes we allocate a connection object to it. When it completed their work then it releases the connection object and this object is added into the pool.

Unit 4

Java 2 Micro Edition(J2ME)

MCQ's

1. All MIDP implementations are required to support what image format?
   a. GIF
   b. JPG
   c. BMP
   d. PNG

Correct answer: d

2. A midlet is a Java class that extends the abstract class?
   a. javax.microedition.MIDlet
   b. javax.microedition.midlet
   c. javax.microedition.midlet.MIDlet
   d. javax.microedition.midlet.midlet

Correct answer: c

3. All MIDP GUI classes are contained in what package?
   a. javax.microedition.gui
   b javax.microedition.lcd
   c. javax.microedition.lcdui
   d. javax.microedition.display

Correct answer: c

4. If a midlet needs to receive high-level events from the implementation, which interface should it implement?
   a. ActionListener
   b. CommandListener
   c. Windows Listener
   d. ChoiceListener

Correct answer: b

5. Which of the following techniques can be used for wireless session tracking?
   a. Cookies
   b. URL Rewriting
   c. Hidden Fields
   d. None of the above

Correct answer: b

6. Which one of the following methods of the MIDlet abstract class must be implemented by a midlet?
   a. initApp, startApp
   b. startApp, destroyApp
   c. initApp, startApp, pauseApp, destroyApp
   d. startApp, pauseApp, destroyApp

Correct answer: d

7. The heart of the Generic Connection framework is?
   a. javax.microedition.Connection
   b. javax.microedition.Connector
   c. javax.microedition.StreamConnectio
   d. javax.microedition.HttpConnection

Correct answer: b

8. Which class would you use to write applications to access low-level input events or to issue graphics calls for drawing to the display?
   a. Wrong
   b. Display
   c. Command
   d. Screen
   e. None of these

Correct answer: d

9. The MIDP user interface API is a subset of the AWT/ Project Swing libraries?
   a. False
   b. True

Correct answer: a

10. A Java virtual machine1 (JVM) supporting CLDC has no support for:

a. Floating point numbers
b. Reflection
c. JNI
d. All of the above

Correct answer: d

Q:1 What is J2ME(Java 2 Micro Edition)?
Ans

J2ME is used to develop tiny devices .
Example: mobile phones, TV set top boxes, Vehicle telematics, pagers, PDAs etc.
Java ME includes :

- flexible user interfaces
- robust security
- built-in network protocols
- support for networked and offline applications that can be downloaded dynamically

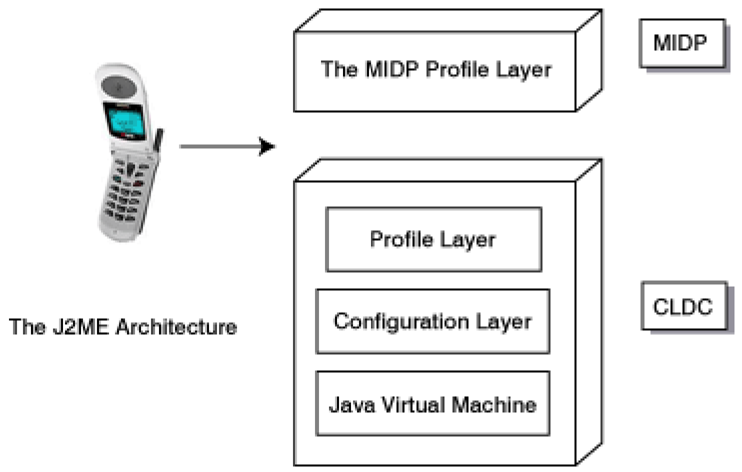
Basically there are two types of configurations involved in Java ME application development, which are:

- CLDC (Connected Limited Device Configuration)
- CDC (Connected Device Configuration)

Q:2 Define J2ME architecture?
Ans

There are 5 layers in **J2ME Architecture.Those are:**

- **MIDP (TopMost Layer):** Which contains Java APIs for user network connections, persistence storage, and the user interface. It also has access to CLDC libraries and MIDP libraries.
- **J2ME API's(Profiles):** Which consists of the minimum set of application programming interfaces for the small computing device
- **Configurations:** Which handles interactions between the profile and the JVM.
- **JVM**
- **Operating System(Bottom Layer).**



J2ME Architecture

Q:3 Define Midlets in J2ME?
Ans

- A **MIDlet** is an **event-based application.**
- A **MIDlet is a J2ME application** which operate on an **MIDP.**
- A **MIDlet** is defined with **at least a single class** that is derived from the **javax.microedition.midlet.MIDlet abstract class.**
- Common programming **is grouping related MIDlets into a MIDlet suite**, which is contained within the same package and implemented simultaneously.
- All **MIDlets within a MIDlet suite** are considered a **group** and must be installed and uninstalled as a group. MIDlets from the same MIDlet suite run the same class.

Three abstract methods are.

- **startApp():** called by the application manager when the MIDlet is started and contains statements that are executed each time the application begins execution. Public and have no return value nor parameter list.

- **pauseApp():** called before the application manager temporarily stops the MIDlet. The application manager restarts the MIDlet by recalling the startApp() method. Public and have no return value nor parameter list.
- **destroyApp():** called prior to the termination of the MIDlet by the application manager. Public method without a return value. It has a boolean parameter that is set to true if the termination of the MIDlet is unconditional, and false if the MIDlet can throw a MIDletStateChangeException.

The Basic Midlet Shell.:

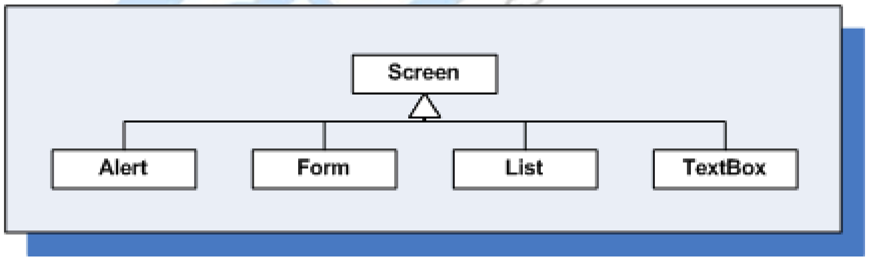
```
public class BasicMIDletShell extends MIDlet
{
public void startApp(){ }
public void pauseApp(){ }
public void destroyApp( boolean unconditional){ }
}
```

Q:4 Define user interface in J2ME?

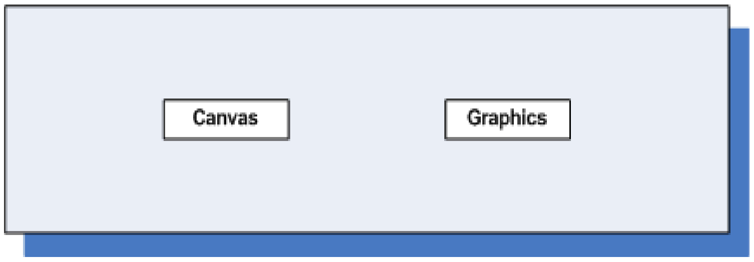
Ans

MIDP 2.0 provides UI classes in two packages:

- javax.microedition.lcdui
- javax.microedition.lcdui.game, where lcdui stands for liquid crystal display user interface (LCD UI).



High-level MIDP 2.0 UI classes



Low-level MIDP 2.0 UI classes

High-level MIDP 2.0 UI classes

1. **Alert**

Alerts are best used in informational or error messages that stay on the screen for a short period of time and then disappear.

2. **List**

A list contains one or more choices (elements), which must have a text part, an optional image part, and an optional font for the text part.

3. **TextBox**

TextBox is a screen's object, using which we can give **input and modify** the text.

4. **Form**

Forms used to **combine multiple components** into one screen. It is a screen that contains items.

Low-level MIDP 2.0 UI classes

1. **Canvas**

This class is used to draw the shapes.

2. Graphics

This class is used to draw the shapes and pictures.

Q:5 Define Event handling with commands in J2ME with example?
Ans

Command Types

NAME	**MEANING**
OK	Confirms a selection.
CANCEL	Cancels pending changes.
BACK	Moves the user back to a previous screen.
STOP	Stops a running operation.
HELP	Shows application instructions.
SCREEN	Generic type for specific application commands.

1. **To create a standard OK command, for example, you would do this:**

   ```
   Command c = new Command("OK", Command.OK, 0);
   ```

2. **To create a command specific to your application, you might do this:**

   ```
   Command c = new Command("Launch", Command.SCREEN, 0);
   ```

3. **You can create a command with a short and long label like this:**

   ```
   Command c = new Command("Run", "Run simulation", Command.SCREEN,
   0);
   ```

4. **The listener is an object that implements the CommandListener interface. To register the listener with a Displayable, use the following method:**

```
public void setListener(CommandListener l)
```

5. **Implementing a CommandListener is a matter of defining a single method:**

```
public void commandAction(Command c, Displayable s)
```

Example:

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class Commander extends MIDlet
{
 public void startApp()
{
 Displayable d = new TextBox("TextBox", "Commander", 20, TextField.ANY);
   Command c = new Command("Exit", Command.EXIT, 0);
 d.addCommand(c);
 d.setCommandListener(new CommandListener() {
   public void commandAction(Command c, Displayable s) {
     notifyDestroyed();
  }
  });

  Display.getDisplay(this).setCurrent(d);
 }

public void pauseApp() { }

 public void destroyApp(boolean unconditional) { }
}
```

Q:6 Define Tickers in J2ME ?
Ans

- A ticker is an object.

- It is associated with the display, not with the screen.
- It used for scrolling text across the top of the display.
- You place a Ticker on a screen using the Screen.setTicker

Constructor :

- **Ticker(String str):-** This is used to Constructs a new Ticker object, given its initial contents string.

Two methods:

- **getString():-** This is the String type methods which Gets the string currently being scrolled by the ticker.
- **setString(String str):-** This is the void type, it Sets the string to be displayed by this ticker.

Q:7 Define Textbox in J2ME ?

Ans

The TextBox class is a Screen .
This allows the user to enter and edit text.

Methods:

- delete(int offset, int length)
- getCaretPosition()
- getChars(char[] data)
- getConstraints()
- getMaxSize()
- getString()
- insert(char[] data, int offset, int length, int position)
- insert(String src, int position)
- setChars(char[] data, int offset, int length)
- setConstraints(int constraints)
- setMaxSize(int maxSize)
- setString(String text)
- size()

Example:

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class TextBoxMIDlet extends MIDlet{
private Display display;
public TextBoxMIDlet()
{
          display = Display.getDisplay(this);
 }
public void startApp()
{
 TextBox t = new TextBox("TextBox Example",

  "Hello Sandeep Kumar Suman !", 256, 0);
          display.setCurrent(t);
  }

  public void pauseApp() {}

  public void destroyApp(boolean unconditional) {}
}
```

Q:8 Define List in J2ME ?

Ans

A list contains one or more choices .

It have a text part, an optional image part, and an optional font for the text part.

Three choices are:

- **EXCLUSIVE**
- **IMPLICIT**
- **MULTIPLE**

Example:

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class ExclusiveChoiceList extends MIDlet{
  private Display display;
  private List list;
  public ExclusiveChoiceList()

 {
         list = new List("Movies", Choice.EXCLUSIVE);
  }

  public void startApp()

 {
  display = Display.getDisplay(this);
  list.append("The Legend of Bhagat Singh", null);
  list.append("Mother India", null);
  list.append("Lagaan", null);
  list.append("Chak De..", null);
  list.append("Hum Aapke Hain Kaun", null);
  display.setCurrent(list);
  }
  public void pauseApp() {}
  public void destroyApp(boolean unconditional){
  notifyDestroyed();
  }
 }
```

Q:9 Define Forms in J2ME ?

Ans

- Form is a screen that contains items.
- **Forms** used to **combine multiple components** into one screen.
- We can also set values into these forms using following syntax:

  public int **append**(Item item)

Example:

```
  import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class AppendItem extends MIDlet{
private Display display;
private MIDlet midlet;
public void startApp()
  {
   Form form = new MyForm("MY FORM");
   display = Display.getDisplay(this);
 display.setCurrent(form);
}

 public void pauseApp(){}
 public void destroyApp(boolean unconditional){}
 class MyForm extends Form implements CommandListener{
 private StringItem textItem;
 private Command exit;
 public MyForm(String title)
  {
 super(title);
 exit = new Command("Exit", Command.BACK, 1);
 this.textItem = new StringItem (null, "Hello Item");
 addCommand(exit);
 append(this.textItem);
 this.setCommandListener(this);
 }

 public void commandAction(Command c, Displayable d){
 String label = c.getLabel();
 if(label.equals("Exit")){
 midlet.notifyDestroyed();
 }
 }
 }
}
```

Unit 5

Java Server Pages

MCQ's

1. The session tracking in the JSP can be done by
    a. URL rewriting
    b. Cookies
    c. User-Authorization
    d. Hidden value

Correct answer: c

2. The jsp:useBean attribute is used to indicate the Serialized bean?
    a. true
    b. false
    c. can't say
    d. none of the above

Correct answer: a

3. What tools can be used to generate the multiple views in jsp ?
    a. xalan-J
    b. Saxon
    c. Jdk1.5
    d. None of the above

Correct answer: a

4. If the session object is invalidated can it be again regained or refreshed?
    a. Yes
    b. No
    c. Can't say
    d. none of the above

Correct answer: b

5. The maximum length of the session Id(length identifier) is ___________?
   a. 4k
   b. 2k
   c. 6k
   d. 8k

Correct answer: a

6. URLConnection instance represents a link for accessing or communicating with the resource at the location?
   a. true
   b. false
   c. can't say
   d. none of the above

Correct answer: a

7. Which of the following statements is true about JSP tag library?
   a. It defines the standard tag that works the same everywhere
   b. It is a single library and we can use it in multiple jsp containers
   c. It has support for the common structural tasks like iteration and condition.
   d. All of the above

Correct answer: d

8. What is the possible way of communication between applet and Servlet ?
   a. HTTP Communication (Text-based and object-based)
   b. Socket Communication
   c. RMI Communication
   d. All of the above

Correct answer: d

9. The listener is notified when the Servlet context (i.e., the Web application) is initialized and destroyed ?
   a. Servlet context attribute
   b. Servlet context listeners
   c. Session Attribute
   d. Session Listeners

Correct answer: b

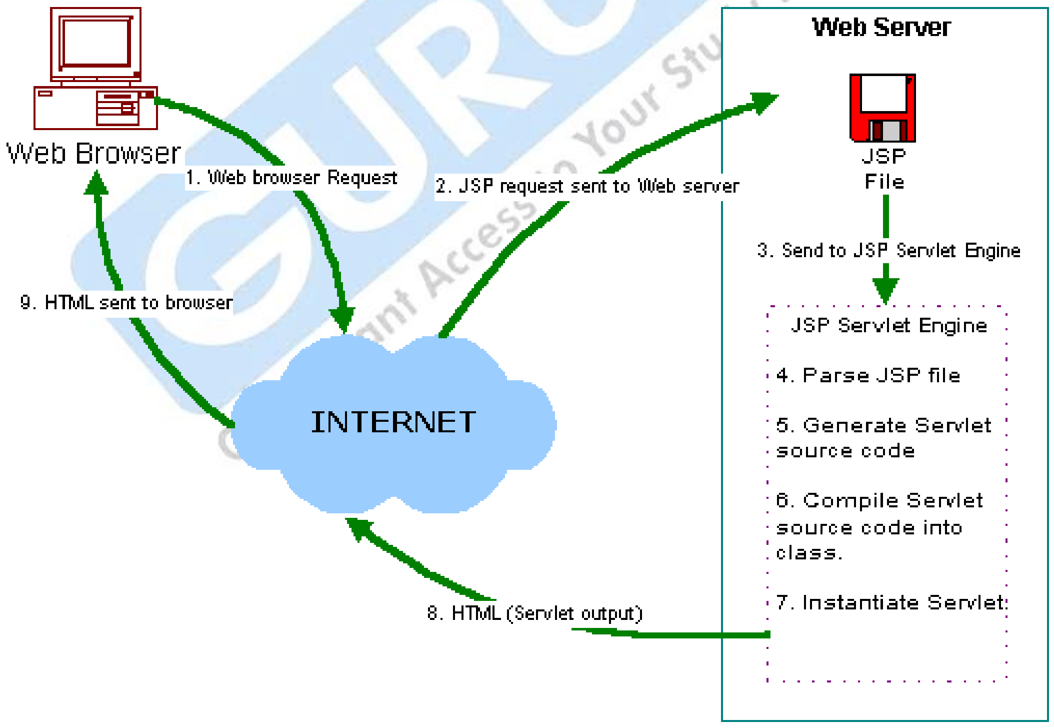
10. The < % return; % > simply aborts the processing of JSP?
    a. true
    b. false
    c. can't say
    d. none of the above

Correct answer: a

Q:1 What is JSP(java server pages) ? Define its architecture?
Ans

- JSP is server side technology.
- JSP technology allows the programmers to embed Java code into html (.jsp) page.
- JSP is used to develop the database driven web applications.
- Java Server Pages are first compiled into Java Servlets by a JSP compiler and then this Servlet is loaded by the Servlet container to server the client request.

Architecture:



Steps:

- The user goes to a web site made using JSP. The user goes to a JSP page (ending with .jsp). The web browser makes the request via the Internet.
- The JSP request gets sent to the Web server.
- The Web server recognises that the file required is special (.jsp),therefore passes the JSP file to the JSP Servlet Engine.
- If the JSP file has been called the first time,the JSP file is parsed,otherwise go to step 7.
- The next step is to generate a special Servlet from the JSP file. All the HTML required is converted to println statements.
- The Servlet source code is compiled into a class.
- The Servlet is instantiated,calling the init and service methods.
- HTML from the Servlet output is sent via the Internet.
- HTML results are displayed on the user's web browser.

Q:2 What are implicit objects in JSP?

Ans Implicit objects are created automatically by the container and are accessed using standard variables.
There are nine implicit objects:

Application: These objects has an application scope. These objects are available at the widest context level, that allows to share the same information between the JSP page's servlet and any Web components with in the same application.

- **Config:** These object has a page scope and is an instance of javax.servlet.ServletConfig class. Config object allows to pass the initialization data to a JSP page's servlet. Parameters of this objects can be set in the deployment descriptor (web.xml) inside the element <jsp-file>. The method getInitParameter() is used to access the initialization parameters.

- **Exception:** This object has a page scope and is an instance of java.lang.Throwable class. This object allows the exception data to be accessed only by designated JSP "error pages."

- **Out:** This object allows us to access the servlet's output stream and has a page scope. Out object is an instance of javax.servlet.jsp.JspWriter class. It provides the output stream that enable access to the servlet's output stream.

- **Page:** This object has a page scope and is an instance of the JSP page's servlet class that processes the current request. Page object represents the current page that is used to call the methods defined by the translated servlet class. First type cast the servlet before accessing any method of the servlet through the page.

- **Pagecontext:** PageContext has a page scope. Pagecontext is the context for the JSP page itself that provides a single API to manage the various scoped attributes. This API is extensively used if we are implementing JSP custom tag handlers. PageContext also provides access to several page attributes like including some static or dynamic resource.

- **Request:** Request object has a request scope that is used to access the HTTP request data, and also provides a context to associate the request-specific data. Request object implements javax.servlet.ServletRequest interface. It uses the getParameter() method to access the request parameter. The container passes this object to the _jspService() method.

- **Response:** This object has a page scope that allows direct access to the **HTTPServletResponse** class object. Response object is an instance of the classes that implements the javax.servlet.ServletResponse class. Container generates to this object and passes to the _jspService() method as a parameter

.

- **Session:** Session object has a session scope that is an instance of javax.servlet.http.HttpSession class. Perhaps it is the most commonly used object to manage the state contexts. This object persist information across multiple user connection.

Q:3 What are standard actions in JSP?

Ans JSP Standard action are predefined tags that are required at the exact time the JSP page is requested by a browser.

They are:

Action element **Description**

<jsp:useBean>	Makes a JavaBeans component available in a page
<jsp:getProperty>	Gets a property value from a JavaBeans component and adds it to the Response
<jsp:setProperty>	Sets a JavaBeans component property value
<jsp:include>	Includes the response from a servlet or JSP page during the request processing phase
<jsp:forward>	Forwards the processing of a request to servlet or JSP page
<jsp:param>	Adds a parameter value to a request handed off to another servlet or JSP page using <jsp:include> or <jsp:forward>
<jsp:plugin>	Generates HTML that contains the appropriate browser-dependent elements (OBJECT or EMBED) needed to execute an applet with the Java Plug-in software

Q:4 What JSP Errors?
Ans

JSP Error Pages specify the custom error page and runtime error
The custom error page helps in handling the error page and display a customized view of the exception.
Two types of error pages:
1) error.jsp
2) errorPage.jsp
1) error.jsp

```
<%@page errorPage="errorPage.jsp" %>
<html>
<head><title>Error Page</title></head>
<body>
<%!
String st = null;
%>
The length of the string is <%= st.length() %>
```

```
        </body>
        </html>

2)      errorPage.jsp
<%@ page isErrorPage="true" import="java.io.*" %>
<html>
<body>
<h2>Error Encountered!</h2>
<font color="red"><%= exception %> has been encountered. You haven't specify
any String.</font>
</body>
</html>
```

Keywords

- **J2EE:** Java Enterprise uses a component based approach for Design, Development,Assembly and Deployment of the enterprise application.

- **Servlet:** Servlets are small java programs used to design the web applications.

- **Thread:**Small program that has single flow of execution.

- **GET():**In this method data is submitted as a part of url.

- **POST():**In this method data is submitted as a part of http request.

- **JDBC: JDBC** is Java application programming interface that allows the Java programmers to access database management system from Java code.

- **Driver:** A **JDBC driver** is a software component enabling a Java application to interact with a database.

- **session:** A Session refers to all the request that a single client makes to a server. A session is specific to the user and for each user a new session is created to track all the request from that user.

- **cookies:** 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.

- **Connection pooling:** Connection pooling is a technique to allow multiple clients to share a cache set of connection objects that provide access to a database resource.

- **JSP:**It is Server side technology used to design web applications.

- **Implicit objects:** Implicit objects are created automatically by the container and are accessed using standard variables.

- **Standard Actions:** JSP Standard action are predefined tags that are required at the exact time the JSP page is requested by a browser.

- **J2ME**:J2ME is used to develop tiny devices .Example: mobile phones, TV set top boxes, Vehicle telematics, pagers, PDAs etc.

- **Midlet**: A **MIDlet** is an **event-based application** of **J2ME application** which operate on an **MIDP.**

- **Config:** These object has a page scope and is an instance of javax.servlet.ServletConfig class. Config object allows to pass the initialization data to a JSP page's servlet.

- **Exception:** This object has a page scope and is an instance of java.lang.Throwable class. This object allows the exception data to be accessed only by designated JSP "error pages."

- **Out:** This object allows us to access the servlet's output stream and has a page scope. Out object is an instance of javax.servlet.jsp.JspWriter class. It provides the output stream that enable access to the servlet's output stream.

- **Page:** This object has a page scope and is an instance of the JSP page's servlet class that processes the current request. Page object represents the current page that is used to call the methods defined by the translated servlet class

- **Pagecontext:** PageContext has a page scope. Pagecontext is the context for the JSP page itself that provides a single API to manage the various scoped attributes.

- **Request:** Request object has a request scope that is used to access the HTTP request data, and also provides a context to associate the request-specific data. Request object implements javax.servlet.ServletRequest interface. It uses the getParameter() method to access the request parameter. The container passes this object to the _jspService() method.

- **Response:** This object has a page scope that allows direct access to the **HTTPServletResponse** class object. Response object is an instance of the classes that implements the javax.servlet.ServletResponse class. Container generates to this object and passes to the jspService() method as a parameter.

- **Session:** Session object has a session scope that is an instance of javax.servlet.http.HttpSession class. Perhaps it is the most commonly used object to manage the state contexts. This object persist information across multiple user connection.

- **Form:Forms** are used to **combine multiple components** into one screen.

- **List:** A list contains one or more choices .

- **Ticker**: It used for scrolling text across the top of the display.

Bibliography

- Harvey M. Deitel, Paul J. Deitel, Sean Santry: *Advanced Java* 2 Platform: How to Program
- Clifford J. Berg, Advanced Java 2: development for enterprise applications
- Gajendra Gupta: *Advance Java*