**UNIT I**

**PART – A (2 Marks)**

1. [**What is J2EE?**](http://www.careerride.com/view.aspx?id=3345)

Java Enterprise Edition is a standard for developing applications based on enterprise softwares. Java Platform, Enterprise Edition or Java EE is an enterprise computing platform on Java from Oracle.

1. **[What are the four components of J2EE application?](http://www.careerride.com/view.aspx?id=3339)**

The four J2EE application components are:

Client tier,

Web tier,

business tier and

enterprise information system (EIS) tier.

1. [**What are the limitation of hibernate?**](http://www.careerride.com/view.aspx?id=3336)

* Performance Cost
* Does not allow multiple inserts
* Poor performance in Batch processing
* Not good for small project

1. **List out the types of J2EE clients?**

* Web (HTML) clients
* Device clients
* Applet clients
* Stand-alone Java clients

1. [**Define Hash table?**](http://www.careerride.com/view.aspx?id=3333)

In computing, a hash table (hash map) is a data structure that implements an associative array abstract data type, a structure that can map keys to values. A hash table uses a hash function to compute an index into an array of buckets or slots, from which the desired value can be found.

1. **Define JDBC?**

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation.

1. **[What are the 3different java platform editions?](http://www.careerride.com/view.aspx?id=3331)**

There are four platforms of the Java programming language:

* Java Platform, Standard Edition (Java SE)
* Java Platform, Enterprise Edition (Java EE)
* Java Platform, Micro Edition (Java ME)
* JavaFX.

1. **[What is URL Encoding?](http://www.careerride.com/view.aspx?id=3328)**

URL encoding is the practice of translating unprintable characters or characters with special meaning within URLs to a representation that is unambiguous and universally accepted by web browsers and servers.

1. [**Explain the lifecycle methods of a servlet?**](http://www.careerride.com/view.aspx?id=3326)

Five steps of servlet life cycle.

Step 1: Loading of Servlet. When the web server (e.g. Apache Tomcat) starts up, the servlet container deploy and loads all the servlets.

Step 2: Creating instance of Servlet. ...

Step 3: Invoke init() method. ...

Step 4: Invoke service() method. ...

Step 5: Invoke destroy() method

1. **[List out the 4 container types in J2EE?](http://www.careerride.com/view.aspx?id=3325)**

* Enterprise JavaBeans (EJB) container
* Web container
* Application client container
* Applet container

**PART – B (16 Marks)**

**1. Explain briefly on the Steps Required to Connect Database using JDBC.**

Step1)At first we need to load the driver into the memory which will be used for connectivity.TheforName() method of the Class class is used for this purpose.

Class.forName("oracle.jdbc.driver.OracleDriver");

This is known as Type4 Driver which is used to connect Oracle Database.

Step2)After the driver is loaded the connection is established by calling the getConnection() method of the DriverManagerclass.It accepts 3 parameter the 1st parameter is URL,2nd Parameter is username and the 3rd Parameter is password. After establishing connection it returns a Connection type object which is stored under the variable of Connection interface.

Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","manager");

Step3)After the connection is established PreparedStatement type object is created to send the query to the database.prepareStatement() method of the Connection interface is used to create PreparedStatement type object.It can also accept input parameter.

PreparedStatementpst=con.prepareStatement("select \* from emp");

Step4)There are 3 methods which are present under the PreparedStatement interface responsible for executing query they are executeQuery(),executeUpdate() and execute().

executeQuery()method is used to execute select statement.

executeUpdate()method is used to execute insert,update and delete statement.

execute()method is used to execute call to a procedure or function.

after the select statement is executed all the rows are stored under the ResultSet type object.

ResultSetrs=pst.executeQuery();

while(rs.next())

{

System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));

}

con.close();

**2.Briefly explain the Steps to manually create a Web Application.**

Steps To Manually create a Web Application

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

step1)At first we need to create a folder say WebApp1.Under WebApp1 create another folder called WEB-INF.

Under WEB-INF create a folder called classes.

step2)Create index.html under WebApp1 and write the following code under it.

<html>

<head>

<title>HomePage</title>

</head>

<body bgcolor="cyan" text="red">

<h1 align="center">Welcome to our HOME PAGE</h1>

<hr size="10" color="green" width="80%">

<h2><a href="MyServlet">Click To Invoke MyServlet</a></h2>

</body>

</html>

step3)Create MyServlet.java under the classes directory present under WEB-INF present under WebApp1.

importjavax.servlet.\*;

importjavax.servlet.http.\*;

import java.io.\*;

public class MyServlet implements Servlet

{

ServletConfigconfig=null;

public void init(ServletConfigconfig)

{

this.config=config;

}

public void service(ServletRequestrequest,ServletResponse response)throws IOException,ServletException

{

response.setContentType("text/html");

PrintWriter out=response.getWriter();

out.println("<h1 align=center>Welcome to our First Servlet</h1>");

out.println("<hr size=10 color=green width=80%>");

}

public void destroy(){}

publicServletConfiggetServletConfig()

{

returnconfig;

}

public String getServletInfo()

{

return null;

}

}

step4)Create deployment descriptor web.xml under WEB-INF folder and write the following code.

<web-app>

<servlet>

<servlet-name>MyServlet</servlet-name>

<servlet-class>MyServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>MyServlet</servlet-name>

<url-pattern>/MyServlet</url-pattern>

</servlet-mapping>

</web-app>

step5)Place servlet-api.jar from lib directory of apache tomcat web server to the classes directory.

step6)Make classes as current directory and type the following command to compile the servlet.

javac -cp .;servlet-api.jar MyServlet.java

step7)Place WebApp1 directory under the webapps directory of apache tomcat web server.and change the port no in server.xml file to 8082.

step8)Start the apache tomcat web server by startup.bat

step9)Open Browser and Type http://localhost:8082/WebApp1

step10)Shutdown the server by shutdown.bat

http://localhost:8082/WebApp2.

**3.Write briefly about the J2ee container centric architecture.**

The JEE platform provides the environment to develop enterprise applications / services using multitier architecture.

- The highly intensified technology made the need for scalable, efficient, faster solutions for information management.

- The JEE technology is rightly apt for meeting these requirements.

- JEE provides a programming development environment which improves the productivity of development, standards for hosting / deploying enterprise applications.

- The following are the tiers in JEE application

**Client Tier :**

- The client tier includes the web components such as Servlets, JSP or standalone Java Desktop applications.

- This tier provides dynamic interfaces to the middle tier.

**Middle Tier:**

- This is also called as the server tier.

- In the middle tier enterprise beans and web services encapsulate distributable business logic for the applications which are reusable.

- The JEE application server contains the server-tier components which provides the platform for these web components for actions to be performed and data to be stored / persisted.

**Enterprise data tier :**

- The enterprise level data is stored / persisted preferably or typically in a relational database.

- In this tier, the JEE applications comprises of components, containers and services.

- All the web components (Servlets, JSP) provide dynamic requests and responses from a web page.

- The EJB components contain the server-side business logic for enterprise applications.

.

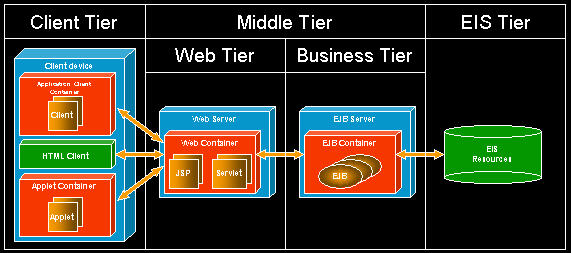
- A container acts as an interface between a platform-specific functionality and a component.

- The component must be assembled before a web or enterprise bean or application client component execution, into a JEE application and deployed into its container.

- The settings of a container can be customized for underlying support provided by the JEE server.

- These include security, transaction management, and Java Naming and Directory Interface lookups.

- The management of non configurable services such as servlet life cycle, enterprise bean life cycle, database connection, data persistence, database connection pooling also can be done by the container.



**four types of container that the J2EE specification defines**.

- A container is a runtime support of a system level entity [Applet, Servlet/JSP, EJB].

- The four types of container that the J2EE specification defines are applet container, application-client container, web container and EJB container.

1. **Applet Container :**

- An applet is a java program that can be embedded into a web page.

- Most of the web pages are authored in HTML.

- To use an applet in HTML document , the tags and are used.

- They are used to indicate to the browser that a java applet should be loaded.

- These tags act as a container for the java applet definition. "Applet container manages the execution of applet, and contains the web browser."

2. **Application-client Container :**

- The Application Client Container (ACC) is a combination of Java classes, libraries, and other files.

- They are used to distribute along with java client programs that execute on their own JVM.

- The execution of the application client components is managed by the application-client container.

- The ACC can take the responsibility to collect user name and password which is treated as authentication data.

**3. The web container :**

- It is used to host web applications. It provides the run time environment to execute Servlet and JSP component types.

**4. EJB container :**

- The business logic is dealt by the server components called EJB components.

- The access to local and remote enterprise beans is provided by the EJB container.

- The operations of the three beans namely Entity Bean, Session Bean and Message-driven bean are handled by the EJB Container**.**

**4.Explain the phases of servlet API.**

A servlet life cycle can be defined as the entire process from its creation till the destruction. The following are the paths followed by a servlet.

The servlet is initialized by calling the init() method.

The servlet calls service() method to process a client's request.

The servlet is terminated by calling the destroy() method.

Finally, servlet is garbage collected by the garbage collector of the JVM.

Now let us discuss the life cycle methods in detail.

**The init() Method**

The init method is called only once. It is called only when the servlet is created, and not called for any user requests afterwards. So, it is used for one-time initializations, just as with the init method of applets.

The servlet is normally created when a user first invokes a URL corresponding to the servlet, but you can also specify that the servlet be loaded when the server is first started.

When a user invokes a servlet, a single instance of each servlet gets created, with each user request resulting in a new thread that is handed off to doGet or doPost as appropriate. The init() method simply creates or loads some data that will be used throughout the life of the servlet.

The init method definition looks like this −

public void init() throws ServletException {

// Initialization code...

}

**The service() Method**

The service() method is the main method to perform the actual task. The servlet container (i.e. web server) calls the service() method to handle requests coming from the client( browsers) and to write the formatted response back to the client.

Each time the server receives a request for a servlet, the server spawns a new thread and calls service. The service() method checks the HTTP request type (GET, POST, PUT, DELETE, etc.) and calls doGet, doPost, doPut, doDelete, etc. methods as appropriate.

Here is the signature of this method −

public void service(ServletRequest request, ServletResponse response)

throwsServletException, IOException {

}

The service () method is called by the container and service method invokes doGet, doPost, doPut, doDelete, etc. methods as appropriate. So you have nothing to do with service() method but you override either doGet() or doPost() depending on what type of request you receive from the client.

The doGet() and doPost() are most frequently used methods with in each service request. Here is the signature of these two methods.

**The doGet() Method**

A GET request results from a normal request for a URL or from an HTML form that has no METHOD specified and it should be handled by doGet() method.

public void doGet(HttpServletRequest request, HttpServletResponse response)

throwsServletException, IOException {

// Servlet code

}

**The doPost() Method**

A POST request results from an HTML form that specifically lists POST as the METHOD and it should be handled by doPost() method.

public void doPost(HttpServletRequest request, HttpServletResponse response)

throwsServletException, IOException {

// Servlet code

}

**The destroy() Method**

The destroy() method is called only once at the end of the life cycle of a servlet. This method gives your servlet a chance to close database connections, halt background threads, write cookie lists or hit counts to disk, and perform other such cleanup activities.

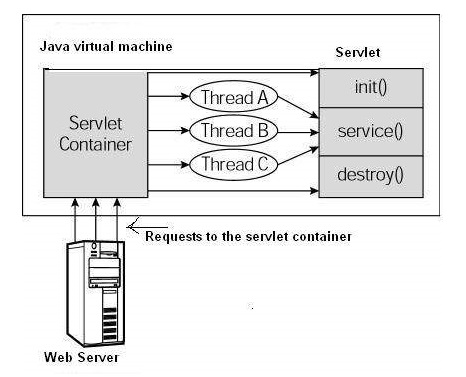
After the destroy() method is called, the servlet object is marked for garbage collection. The destroy method definition looks like this −

public void destroy() {

// Finalization code...

}

**Architecture Diagram**



**5.Discuss briefly on the tiers of the j2ee application.(refer qn no:3)**

**UNIT – II**

**PART – A (2 Marks)**

1. Define JSP?

java Server Pages technology (JSP) is a server-side programming language used to create a dynamic web page in the form of HyperText Markup Language (HTML). It is an extension to the servlet technology.How to call a jsp page from a servlet?

1. What is a deployment descriptor?

A deployment descriptor (DD) refers to a configuration file for an artifact that is deployed to some container/engine. In the Java Platform, Enterprise Edition, a deployment descriptor describes how a component, module or application (such as a web application or enterprise application) should be deployed.

1. Difference between servlet and JSP

Only one object at the time of first request by servlet or web container.

.java Server Pages technology (JSP) is a server-side programming language used to create a dynamic web page in the form of HyperText Markup Language (HTML).

1. Define jsp tags.

In JSP, java code can be written inside the jsp page using the scriptlet tag.

1. What is the syntax of JSP?

<% code fragment %>

<jsp:scriptlet>

code fragment

</jsp:scriptlet>

1. List out the two Advantages of JSP over Servlets? .

Servlets are difficult to code which are overcome in JSP. Other way, we can say, JSP is almost a replacement of Servlets, (by large, the better word is extension of Servlets), where coding decreases more than half.

In Servlets, both static code and dynamic code are put together. In JSP, they are separated.

1. List out any 4 JSP action tags?

|  |  |
| --- | --- |
| JSP Action | Description |
| jsp:element | To define the XML elements dynamically. |
| jsp:attribute | To define the dynamically generated XML element attributes |
| jsp:body | To define the dynamically generated XML element body |
| jsp:plugin | To generate the browser-specific code that makes an OBJECT or EMBED tag for the Java plugin. |
|  |  |

8.How to call a jsp page from a servlet?

<jsp:include page="/servlet/MyServlet" flush="true" />

<jsp:forward page="/servlet/MyServlet" />

As when invoking one JSP page from another, you can invoke a servlet from a JSP page through the jsp:include and jsp:forward action tags.

9.What are the life cycle methods of JSP?

|  |  |
| --- | --- |
| **Method** | **Description** |
| public void jspInit() | It is invoked only once, same as init method of the servlet. |
| public void \_jspService(ServletRequestrequest,ServletResponse)throws ServletException,IOException | It is invoked at each request, same as service() method of the servlet. |
| public void jspDestroy() | It is invoked only once, same as destroy() method of the servlet. |

10.what is JSP directive?

These directives provide directions and instructions to the container, telling it how to handle certain aspects of the JSP processing.

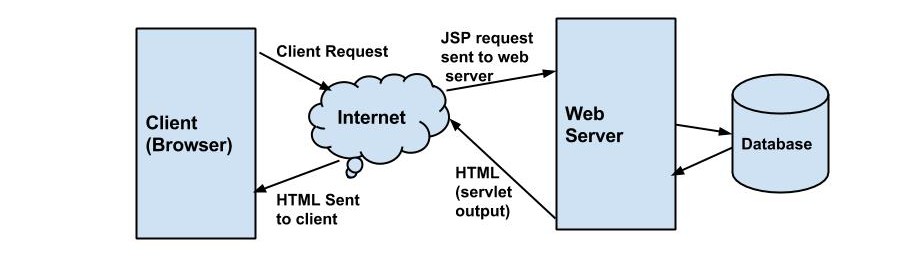
**PART – B (16 Marks)**

**1.Briefly explain the JSP architecture**?

The web server needs a JSP engine ie. container to process JSP pages. The JSP container is

responsible for intercepting requests for JSP pages.

A JSP container works with the Web server to provide the runtime environment and other servicesa JSP needs.



**JSP Processing:**

The following steps explain how the web server creates the web page using JSP:

* As with a normal page, your browser sends an HTTP request to the web server.
* The web server recognizes that the HTTP request is for a JSP page and forwards it to a JSP

engine.

* This is done by using the URL or JSP page which ends with .jsp instead of .html.
* The JSP engine loads the JSP page from disk and converts it into a servlet content. This

conversion is very simple in which all template text is converted to println statements and all

JSP elements are converted to Java code that implements the corresponding dynamic

behavior of the page.

* The JSP engine compiles the servlet into an executable class and forwards the original

request to a servlet engine.

* A part of the web server called the servlet engine loads the Servlet class and executes it.
* During execution, the servlet produces an output in HTML format, which the servlet engine

passes to the web server inside an HTTP response.

* The web server forwards the HTTP response to your browser in terms of static HTML content.
* Finally web browser handles the dynamically generated HTML page inside the HTTP

response exactly as if it were a static page.

**2.Discuss in detail about the paths followed by JSP?**

The following are the paths followed by a JSP

Compilation

Initialization

Execution

Cleanup

**JSP Compilation:**

When a browser asks for a JSP, the JSP engine first checks to see whether it needs to compile the page. If the page has never been compiled, or if the JSP has been modified since it was last compiled, the JSP engine compiles the page.

The compilation process involves three steps:

Parsing the JSP.

Turning the JSP into a servlet.

Compiling the servlet.

**JSP Initialization:**

When a container loads a JSP it invokes the jspInit() method before servicing any requests. If you need to perform JSP-specific initialization, override the jspInit() method:

public void jspInit(){ // Initialization code... }

Typically initialization is performed only once and as with the servlet init method, you generally initialize database connections, open files, and create lookup tables in the jspInit method.

**JSP Execution:**

This phase of the JSP life cycle represents all interactions with requests until the JSP is destroyed.

Whenever a browser requests a JSP and the page has been loaded and initialized, the JSP engine invokes the \_jspService() method in the JSP.

The \_jspService() method takes an HttpServletRequest and an HttpServletResponse as its parameters as follows:

void \_jspService(HttpServletRequest request,

HttpServletResponse response)

{

// Service handling code...0

}

The \_jspService() method of a JSP is invoked once per a request and is responsible for generating the response for that request and this method is also responsible for generating responses to all seven of the HTTP methods ie. GET, POST, DELETE etc.

**JSP Cleanup:**

The destruction phase of the JSP life cycle represents when a JSP is being removed from use by a container.

The jspDestroy() method is the JSP equivalent of the destroy method for servlets. Override jspDestroy when you need to perform any cleanup, such as releasing database connections or closing open files.

The jspDestroy() method has the following form:

public void jspDestroy()

{

// Your cleanup code goes here.

}

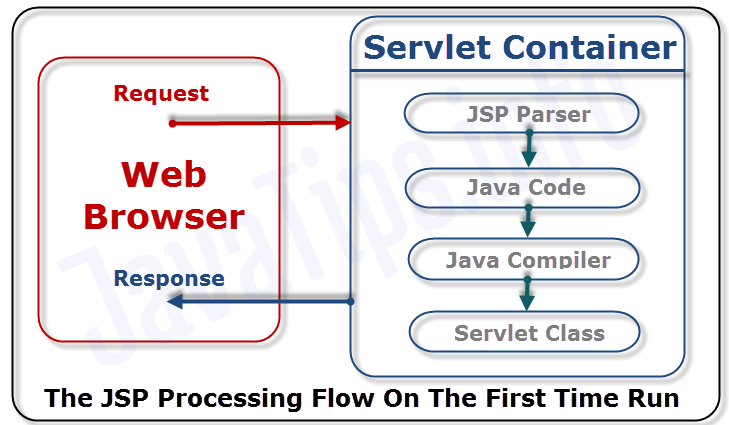
**3.Illustrate in detail the steps involved in web application debugging.**

To enable debugging

* Access the Server Manager and click the Java tab.
* Click JVM General.
* Specify the Java home in the Java Home field.
* The Java home is the path to the directory where the JDK is installed.
* To enable debugging, select On from the Debug Enabled drop-down list
* Specify debug options in the Debug Options field.
* Click OK.

4.Explain how the JSP page works?

JavaServer Pages are made operable by having their contents (HTML tags, JSP tags and scripts) translated into a servlet by the application server.

[](http://javatips.info/wp-content/uploads/2015/12/JSPLifeCycle1stJT_wm.png)

1. A request is sent to the JSP page from web browser.
2. The content of the JSP file is parsed by the JSP engine.
3. The JSP engine translates the JSP to  a servlet java file based on the content of the JSP. The generated servlet is responsible for rendering the static elements of the JSP specified at design time in addition to creating the dynamic elements of the page.
4. The servlet java file is compiled by the Java compiler into a servlet class file.
5. The servlet is instantiated. The init and service methods of the servlet are called, and the servlet logic is executed.
6. The static HTML, graphics and dynamic elements specified in the original JSP page definition are sent to the Web browser through the output stream of the servlet’s response object.

**5.Write a Simple JSP program with Eclipse and Tomcat.**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <%@ page language="java" contentType="text/html; charset=ISO-8859-1"      pageEncoding="ISO-8859-1"%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "<a class="vglnk" href="<http://www.w3.org/TR/html4/loose.dtd>" rel="nofollow"><span>http</span><span>://</span><span>www</span><span>.</span><span>w3</span><span>.</span><span>org</span><span>/</span><span>TR</span><span>/</span><span>html4</span><span>/</span><span>loose</span><span>.</span><span>dtd</span></a>">  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">  <title>Insert title here</title>  </head>  <body>   <%= "Hello World!" %>  </body>  </html> |