1. What gets printed when the following expression is evaluated? Select the one correct answer.**${(1==2) ? 4 : 5}**
    1. 1
    2. 2
    3. 4
    4. 5
2. What gets printed when the following expression is evaluated? Select the one correct answer.**${4 div 5}**
    1. 0
    2. 0.8
    3. 1
    4. -1
3. What gets printed when the following expression is evaluated? Select the one correct answer.**${12 % 4}**
    1. 0
    2. 3
    3. 8
    4. 16
4. What is the effect of executing the following JSP statement, assuming a class with name Employee exists in classes package.

```
<%@ page import = “classes.Employee” %> <jsp:useBean
id=”employee” class=”classes.Employee” scope=”session”/>
<jsp:setProperty name=”employee” property=”*”/>
```

    1. The code does not compile as there is no property attribute of setProperty tag.
    2. The code does not compile as property attribute cannot take * as a value.
    3. The code sets value of all properties of employee bean to “*”.
    4. The code sets the values of all properties of employee bean to matrching parameters in request object.
5. What is the effect of evaluation of following expression? Select the one correct answer. **${(5*5) ne 25}**
    1. true
    2. false
    3. 25
    4. The expression does not compile as ne is not a valid operator.
6. What is the effect of evaluation of following expression? Select the one correct answer. **${‘cat’ gt ‘cap’}**
    1. true
    2. false
    3. catcap
    4. The expression does not compile as gt operator cannot be applied on strings.
7. How many numbers are printed, when the following JSTL code fragment is executed? Select the one correct answer.

```
<%@ taglib uri=”http://java.sun.com/jsp/jstl/core” prefix=”c” %>
<c:forEach var=”item” begin=”0″ end=”10″ step=”2″>
${item}
</c:forEach>
```

    1. 1
    2. 5
    3. 6
    4. 11
8. What gets printed when the following JSTL code fragment is executed? Select the one correct answer.

```
<%@ taglib uri=”http://java.sun.com/jsp/jstl/core” prefix=”c” %>
<c:set var=”item” value=”2″/>
<c:if test=”${var==1}” var=”result” scope=”session”>
<c:out value=”${result}”/>
```

```
</c:if>
```

1. The JSTL code does not compile as attribute for if tag are not correct.
2. true
3. false
4. Nothing gets printed.

9. What gets printed when the following JSTL code fragment is executed? Select the one correct answer.

```
<%@ taglib uri=”http://java.sun.com/jsp/jstl/core” prefix=”c” %>
<c:set var=”item” value=”2″/>
<c:forEach var=”item” begin=”0″ end=”0″ step=”2″>
<c:out value=”${item}” default=”abc”/>
</c:forEach>
```

1. The JSTL code does not compile as an attribute for forEach tag is not correct.
2. 0
3. 2
4. ABC
5. Nothing gets printed as c.out statement does not get executed.

10. How many numbers gets printed when the following JSTL code fragment is executed? Select the one correct answer.

```
<%@ taglib uri=”http://java.sun.com/jsp/jstl/core” prefix=”c” %>
<c:set var=”item” value=”2″/>
<c:choose>
<c:when test=”${item>0}”>
<c:out value=”1″/>
</c:when>
<c:when test=”${item==2}”>
<c:out value=”2″/>
</c:when>
<c:when test=”${item<2}”>
<c:out value=”3″/>
</c:when>
<c:otherwise>
<c:out value=”4″/>
</c:otherwise>
</c:choose>
```

1. No number gets printed.
2. One number gets printed.
3. Two numbers gets printed.
4. Three numbers gets printed.
5. Four numbers gets printed.

11. Which numbers gets printed when the following JSTL code fragment is executed? Select the two correct answers.

```
<%@ taglib uri=”http://java.sun.com/jsp/jstl/core” prefix=”c” %>
<c:set var=”j” value=”4,3,2,1″/>
<c:forEach items=”${j}” var=”item” begin=”1″ end=”2″>
<c:out value=”${item}” default=”abc”/>
</c:forEach>
```

1. 1

2. 2
3. 3
4. 4
5. abc
6. The program does not compile.

12. Which numbers gets printed when the following JSTL code fragment is executed? Select the two correct answers.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:set var="j" value="4,3,2,1″/>
<c:forEach items="${j}" var="item" begin="1″ end="2″
varStatus="status">
<c:out value="${status.count}" default="abc"/>
</c:forEach>
```

1. 1
2. 2
3. 3
4. 4
5. abc
6. The program does not compile.

13. Which number gets printed when the following JSTL code fragment is executed? Select the one correct answers.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:set var="j" value="4,3,2,1″/>
<c:forEach items="${j}" var="item" varStatus="status">
<c:if test="${status.first}">
<c:out value="${status.index}" default="abc"/>
</c:if>
</c:forEach>
```

1. 1
2. 2
3. 3
4. 4
5. abc
6. The program does not compile.

14. Which of these represent the correct path for the core JSTL library in JSTL version 1.1? Select the one correct answer.
    1. http://java.sun.com/jsp/jstl/core
    2. http://java.sun.com/jsp/core
    3. http://java.sun.com/core
    4. http://java.sun.com/jsp/jstl1.1/core

Answers to questions on EL and JSTL

1. D. As 1 is not equal to 2, 5 gets printed.
2. B. div operator is used for dividing in EL.
3. A. % operator gives the remainder after performing division.
4. D. This is a valid syntax for **setProperty**. All properties of the bean are set from the corresponding parameter names in the request object.
5. B. The code prints false. ne is a valid operator. Since both left hand side and right hand side are equal to 25, false gets printed.
6. A. EL considers <cat> to be greater than <cap>, as the letter t comes after the letter p.

7. C. The following numbers get printed – 0, 2, 4, 6, 8, 10.
8. D. if evaluates to false, hence the **c.out** statement does not get executed.
9. B. The **forEach** tag gets executed once, and prints zero.
10. B. Only one number gets printed – the number 1.
11. B, C. In this case the forEach tag iterates through two elements of the array named j.
12. B, C. varStatus is set to a class of type LoopTagStatus. This class has a property named count which is being printed. count is the loop index, beginning with 1. So for two iterations 1 and 2 get printed. In this case the **forEach** tag iterates through two elements of the array named j.
13. A. status.first is true for the first iteration. The index is set to 0 in the first iteration.
14. A. The path of core tag library in JSTL 1.1 is http://java.sun.com/jsp/jstl/core .

Test

1)What is dirty checking in Hibernate?

A. ✖ Object state changes in order to synchronize the updated state with the database

B. Remove the dirty data from data base.

C. Check the data when insert into data base.

D. None

Answer

A :

Object state changes in order to synchronize the updated state with the database.

Dirty Checking is one of the features of hibernate. In dirty checking, hibernate automatically detects whether an object is modified (or) not and need to be updated. As long as the object is in persistent state i.e., bound to a particular Session(org.hibernate.Session). Hibernate monitors any changes to the objects and executes sql.

2)what does hibernate.hbm2ddl.auto create this means?

A. ✔ Create tables automatically

B. Create session object automatically

C. Create Session Factory object automatically

D. None

Answer

A :

Create tables automatically

hibernate.hbm2ddl.auto Automatically validates or exports schema DDL to the database when the SessionFactory is created. With create-drop, the database schema will be dropped when the SessionFactory is closed explicitly.
e.g. validate | update | create | create-drop
So the list of possible options are,

1. validate: validate the schema, makes no changes to the database.
2. update: update the schema.
3. create: creates the schema, destroying previous data.
4. create-drop: drop the schema at the end of the session.

⚠ Not Attempted

3) Which statement is correct about session.load()?

A. load() will throw an unrecoverable exception if there is no matching database row.

B. load() will return null if there is no matching database row.

C. NA

D. NA

Answer

A :

load() will throw an unrecoverable exception if there is no matching database row.

session.load() :-
It will always return a â€œproxyâ€ (Hibernate term) without hitting the database. In Hibernate, proxy is an object with the given identifier value, its properties are not initialized yet, it just look like a temporary fake object.
If no row found , it will throws an ObjectNotFoundException.

4)Is Hibernate Session thread safe?

A. ✖ Yes.

B. No.

C. No relation with thread .

D. None.

Answer

B :

No

Session is a light weight and a non-threadsafe object (No, you cannot share it between threads) that represents a single unit-of-work with the database. Sessions are opened by a SessionFactory and then are closed when all work is complete. Session is the primary interface for the persistence service.

⚠ Not Attempted

5) What does session.createCriteria().uniqueResult() return?

A. Object

B. String

C. ResultSet

D. HibernateResultSet

Answer

A :

Object

Hibernate Criteria class is available in org.hibernate.criteria package. The uniqueResult() method returns single instance of a persistence object. If no result found then method will return a null value.
Syntax :

```
Criteria criteria = session.createCriteria(StudentBean.class)
.add(Restrictions.eq(propertyName,propertyValue));
modelClass obModel = (modelClass ) criteria.uniqueResult();)
```

⚠ Not Attempted

6) How can get a session object ?

A. SessionFactory.getSession();

B. SessionFactory.openSession();

C. SessionFactory.get();

D. (session)SessionFactory.getObject();

Answer

A :

SessionFactory.getSession();

public static org.hibernate.Session getSession(org.hibernate.SessionFactory sessionFactory,boolean allowCreate) throws DataAccessResourceFailureException, IllegalStateException Get a Hibernate Session for the given SessionFactory. Is aware of and will return any existing corresponding Session bound to the current thread, for example when using HibernateTransactionManager. Will create a new Session otherwise, if ";allowCreate"; is true.
This is the getSession method used by typical data access code, in combination with releaseSession called when done with the Session. Note that HibernateTemplate allows to write data access code without caring about such resource handling.
Parameters:
1. sessionFactory - Hibernate SessionFactory to create the session with
2. allowCreate - whether a non-transactional Session should be created when no transactional Session can be found for the current thread
Returns:
the Hibernate Session
Throws:
1. DataAccessResourceFailureException - if the Session couldn't be created
2. IllegalStateException - if no thread-bound Session found and ";allowCreate"; is false

⚠ Not Attempted

7) What is the root level element in a hibernate mapping file?

A. <hibernate-mapping>

B. <session-mapping>

C. <sessionfactory-mapping>

D. none of the above

Answer

A :

<hibernate-mapping>

<hibernate-mapping> element
The first or root element of hibernate mapping document is <hibernate-mapping> element.
Between the <hibernate-mapping> tag class element(s) are present.
The general structure is as follows:-

```
<hibernate-mapping>
  <class name="roseindia.tutorial.hibernate.Contact" table="CONTACT">
   <id name="id" type="long" column="ID" >
   <generator class="assigned"/>
  </id>

  <property name="firstName">
   <column name="FIRSTNAME" />
  </property>
  <property name="lastName">
  <column name="LASTNAME"/>
  </property>
  <property name="email">
  <column name="EMAIL"/>
  </property>
 </class>
</hibernate-mapping>
```

⚠ Not Attempted

8) Is this configuration correct in hibernate?
<property name="askby" column="askby" type="string" length="200"/>
<property name="askby" column="brief" type="string" length="500"/>

A. Yes

B. No

C. Both 1 and 2

D. None

Answer

B :

No

Column parameter of property defines the name of the mapped database table column.
In above case askby and brief database table columns are mapped to same property name askby which creates ambiguity so this is incorrect configuration in hibernate.

⚠ Not Attempted

9) What does CacheMode do?

A. Controls how a particular SessionFactory interacts with the Data Base.

B. Controls how a particular SessionFactory interacts with the second-level cache.

C. Controls how a particular session interacts with the second-level cache.

D. None of this

Answer

⚠ Not Attempted

10) Which statement is correct?

A. session.contains() method to determine if an instance belongs to the session cache.

B. session.contains() method to determine if an instance belongs to the data base.

C. Both are correct

D. None of the above

Answer

A :

session.contains() method to determine if an instance belongs to the session cache.

The Session also provides a contains() method to determine if an instance belongs to the session cache.

Restart Test (session/logout.php)

****

www.techfaq360.com
Hibernate Interview Questions
****
Question No : 1
Difference between session.save() , session.saveOrUpdate() and session.persist()?
Answer :
session.save() : Save does an insert and will fail if the primary key is already persistent.

session.saveOrUpdate() : saveOrUpdate does a select first to determine if it needs to do an insert or an update.
Insert data if primary key not exist otherwise update data.

session.persist() : Does the same like session.save().
But session.save() return Serializable object but session.persist() return void.
session.save() returns the generated identifier (Serializable object) and session.persist() doesn't.
For Example :
if you do :-
System.out.println(session.save(question));
This will print the generated primary key.
if you do :-
System.out.println(session.persist(question));
Compile time error because session.persist() return void.

Question No : 2
Q.What is the difference between hibernate and jdbc ?
Answer :
There are so many

1) Hibernate is data base independent, your code will work for all ORACLE,MySQL ,SQLServer etc.

In case of JDBC query must be data base specific.

2) As Hibernate is set of Objects , you don?t need to learn SQL language.

You can treat TABLE as a Object . Only Java knowledge is need.

In case of JDBC you need to learn SQL.

3) Don?t need Query tuning in case of Hibernate. If you use Criteria Quires in Hibernate then hibernate automatically tuned your query and return best result with performance.

In case of JDBC you need to tune your queries.

4) You will get benefit of Cache. Hibernate support two level of cache. First level and 2nd level. So you can store your data into Cache for better performance.

In case of JDBC you need to implement your java cache .

5) Hibernate supports Query cache and It will provide the statistics about your query and database status.

JDBC Not provides any statistics.

6) Development fast in case of Hibernate because you don?t need to write queries

7) No need to create any connection pool in case of Hibernate. You can use c3p0.

In case of JDBC you need to write your own connection pool

8) In the xml file you can see all the relations between tables in case of Hibernate. Easy readability.

9) You can load your objects on start up using lazy=false in case of Hibernate.

JDBC Don?t have such support.

10) Hibernate Supports automatic versioning of rows but JDBC Not.

Question No : 3

Q. What is lazy fetching in Hibernate? With Example .

Answer :

Lazy fetching decides whether to load child objects while loading the Parent Object.

You need to do this setting respective hibernate mapping file of the parent class.

Lazy = true (means not to load child)

By default the lazy loading of the child objects is true.

This make sure that the child objects are not loaded unless they are explicitly invoked in the application by calling getChild() method on parent.In this case hibernate issues a fresh database call to load the child when getChild() is actully called on the Parent object
.But in some cases you do need to load the child objects when parent is loaded.
Just make the lazy=false and hibernate will load the child when parent is loaded from the database.
Example :
If you have a TABLE ? EMPLOYEE mapped to Employee object and contains set of Address objects.
Parent Class : Employee class
Child class : Address Class

```
public class Employee {
private Set address = new HashSet(); // contains set of child Address objects
    public Set getAddress () {
        return address;
    }
    public void setAddresss(Set address) {
        this. address = address;
    }
}
```

In the Employee.hbm.xml file

```
<set name="address" inverse="true" cascade="delete" lazy="false">
                    <key column="a_id" />
            <one-to-many  class="beans Address"/>
</set>
```

In the above configuration.
If lazy="false" : - when you load the Employee object that time child object Adress is also loaded and set to setAddresss() method.
If you call employee.getAdress() then loaded data returns.No fresh database call.

If lazy="true" :- This the default configuration. If you don?t mention then hibernate consider lazy=true.
when you load the Employee object that time child object Adress is not loaded. You need extra call to data base to get address objects.
If you call employee.getAdress() then that time database query fires and return results. Fresh database call.

Question No : 4
Q.what is the advantage of Hibernate over jdbc?
Answer :
There are so many

1) Hibernate is data base independent, your code will work for all ORACLE,MySQL ,SQLServer etc.
In case of JDBC query must be data base specific.
2) As Hibernate is set of Objects , you don?t need to learn SQL language.
You can treat TABLE as a Object . Only Java knowledge is need.
In case of JDBC you need to learn SQL.
3) Don?t need Query tuning in case of Hibernate. If you use Criteria Quires in Hibernate then hibernate automatically tuned your query and return best result with performance.
In case of JDBC you need to tune your queries.
4) You will get benefit of Cache. Hibernate support two level of cache. First level and 2nd level. So you can store your data into Cache for better performance.
In case of JDBC you need to implement your java cache .

5) Hibernate supports Query cache and It will provide the statistics about your query and database status.
JDBC Not provides any statistics.
6) Development fast in case of Hibernate because you don?t need to write queries
7) No need to create any connection pool in case of Hibernate. You can use c3p0.
In case of JDBC you need to write your own connection pool
8) In the xml file you can see all the relations between tables in case of Hibernate. Easy readability.
9) You can load your objects on start up using lazy=false in case of Hibernate.
JDBC Don?t have such support.

10) Hibernate Supports automatic versioning of rows but JDBC Not.

Question No : 5
How to Integrate Struts Spring Hibernate ?
Answer :
Details with code you can deploy in tomcat server and test .

[URL=http://www.techfaq360.com/tutorial/spring/struts_spring_hibernate.jsp]http://www.techfaq360.com/tutorial/spring/struts_spring_hibernate.jsp[/URL]

Step 1.
In the struts-config.xml add plugin

```
<plug-in className="org.springframework.web.struts.ContextLoaderPlugIn">
        <set-property property="contextConfigLocation"
           value="/WEB-INF/applicationContext.xml"/>
    </plug-in>
```

Step 2.

In the applicationContext.xml file
Configure datasourse

```
    <bean id="dataSource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
     <property name="driverClassName"><value>oracle.jdbc.driver.OracleDriver</value>
      </property>
        <property name="url"><value>jdbc:oracle:thin:@10.10.01.24:1541:ebizd</value>
        </property>
        <property name="username"><value>sa</value></property>
   <property name="password"><value></value></property>
    </bean>
```

Step 3.

Configure SessionFactory

```
 <!-- Hibernate SessionFactory -->
    <bean id="sessionFactory"
class="org.springframework.orm.hibernate.LocalSessionFactoryBean">
        <property name="dataSource"><ref local="dataSource"/></property>
        <property name="mappingResources">
            <list>
                <value>com/test/dbxml/User.hbm.xml</value>
            </list>
        </property>
        <property name="hibernateProperties">
        <props>
            <prop key="hibernate.dialect">net.sf.hibernate.dialect.OracleDialect </prop>
         </props>
        </property>
    </bean>
```

Step 4.
Configure User.hbm.xml

```
<hibernate-mapping>
    <class name="org.test.model.User" table="app_user">

        <id name="id" column="id" >
            <generator class="increment"/>
```

```
        </id>
        <property name="firstName" column="first_name" not-null="true"/>
        <property name="lastName" column="last_name" not-null="true"/>

    </class>
</hibernate-mapping>
```

Step 5.

In the applicationContext.xml ? configure for DAO

```
 <bean id="userDAO" class="org.test.dao.hibernate.UserDAOHibernate">
        <property name="sessionFactory"><ref local="sessionFactory"/></property>
    </bean>
```

Step 6.

DAO Class

```
public class UserDAOHibernate extends HibernateDaoSupport implements UserDAO {
    private static Log log = LogFactory.getLog(UserDAOHibernate.class);

    public List getUsers() {
        return getHibernateTemplate().find("from User");
    }

    public User getUser(Long id) {
        return (User) getHibernateTemplate().get(User.class, id);
    }

    public void saveUser(User user) {
        getHibernateTemplate().saveOrUpdate(user);

        if (log.isDebugEnabled()) {
            log.debug("userId set to: " + user.getId());
        }
    }

    public void removeUser(Long id) {
        Object user = getHibernateTemplate().load(User.class, id);
        getHibernateTemplate().delete(user);
    }
```

}

Question No : 6
How to prevent concurrent update in Hibernate?
Answer :
 version checking used in hibernate when more then one thread trying to access same data.
  For example :
  User A edit the row of the TABLE for update (In the User Interface changing data - This is user thinking time)
  and in the same time User B edit the same record for update and click the update.
  Then User A click the Update and update done. Chnage made by user B is gone.

  In hibernate you can perevent slate object updatation using version checking.

  Check the version of the row when you are upding the row.
  Get the version of the row when you are fetching the row of the TABLE for update.
  On the time of updation just fetch the version number and match with your version number (on the time of fetching).

  This way you can prevent slate object updatation.

  Steps 1:
  Declare a variable "versionId" in your Class with setter and getter.
  public class Campign {
  private Long versionId;
  private Long campignId;
  private String name;
  public Long getVersionId() {
  return versionId;
  }
  public void setVersionId(Long versionId) {
  this.versionId = versionId;
  }

  public String getName() {
  return name;
  }
  public void setName(String name) {
  this.name = name;
  }

```
 public Long getCampignId() {
 return campignId;
}
private void setCampignId(Long campignId) {
 this.campignId = campignId;
}

 }
```

Step 2.
In the .hbm.xml file

```
<class name="beans.Campign" table="CAMPIGN" optimistic-lock="version">

      <id name="campignId" type="long" column="cid">
      <generator class="sequence">
            <param name="sequence">CAMPIGN_ID_SEQ</param>

      </generator>
   </id>
<version name="versionId" type="long" column="version" />

      <property name="name" column="c_name"/>

   </class>
```

Step 3.
Create a coulmn name "version" in the CAMPIGN table.

Step 4.
In the code

```
// foo is an instance loaded by a previous Session
session = sf.openSession();
int oldVersion = foo.getVersion();
session.load( foo, foo.getKey() );
if ( oldVersion!=foo.getVersion ) throw new StaleObjectStateException();
foo.setProperty("bar");
session.flush();
session.connection().commit();
session.close();
```

You can handle StaleObjectStateException() and do what ever you want.

You can display error message.

Hibernate autumatically create/update the version number when you update/insert any row in the table.

Question No : 7

How to perevent slate object updatation in Hibernate ?

Answer :

version checking used in hibernate when more then one thread trying to access same data.

For example :

User A edit the row of the TABLE for update (In the User Interface changing data - This is user thinking time)

and in the same time User B edit the same record for update and click the update.

Then User A click the Update and update done. Chnage made by user B is gone.

In hibernate you can perevent slate object updatation using version checking.

Check the version of the row when you are upding the row.

Get the version of the row when you are fetching the row of the TABLE for update.

On the time of updation just fetch the version number and match with your version number (on the time of fetching).

This way you can prevent slate object updatation.

Steps 1:

Declare a variable "versionId" in your Class with setter and getter.

```
public class Campign {
private Long versionId;
private Long campignId;
private String name;
public Long getVersionId() {
return versionId;
}
public void setVersionId(Long versionId) {
this.versionId = versionId;
}

public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
```

```
 }

 public Long getCampignId() {
 return campignId;
}
private void setCampignId(Long campignId) {
 this.campignId = campignId;
}

 }
```

Step 2.
In the .hbm.xml file

```
<class name="beans.Campign" table="CAMPIGN" optimistic-lock="version">

      <id name="campignId" type="long" column="cid">
      <generator class="sequence">
            <param name="sequence">CAMPIGN_ID_SEQ</param>

      </generator>
   </id>
<version name="versionId" type="long" column="version" />

      <property name="name" column="c_name"/>

   </class>
```

Step 3.
Create a coulmn name "version" in the CAMPIGN table.

Step 4.
In the code

```
// foo is an instance loaded by a previous Session
session = sf.openSession();
int oldVersion = foo.getVersion();
session.load( foo, foo.getKey() );
if ( oldVersion!=foo.getVersion ) throw new StaleObjectStateException();
foo.setProperty("bar");
session.flush();
session.connection().commit();
session.close();
```

You can handle StaleObjectStateException() and do what ever you want.
You can display error message.

Hibernate autumatically create/update the version number when you update/insert any row in the table.

Question No : 8
What is version checking in Hibernate ?
Answer :
version checking used in hibernate when more then one thread trying to access same data.
For example :
User A edit the row of the TABLE for update (In the User Interface changing data - This is user thinking time)
and in the same time User B edit the same record for update and click the update.
Then User A click the Update and update done. Chnage made by user B is gone.

In hibernate you can perevent slate object updatation using version checking.

Check the version of the row when you are upding the row.
Get the version of the row when you are fetching the row of the TABLE for update.
On the time of updation just fetch the version number and match with your version number (on the time of fetching).

This way you can prevent slate object updatation.

Steps 1:
Declare a variable "versionId" in your Class with setter and getter.

```
public class Campign {
private Long versionId;
private Long campignId;
private String name;
public Long getVersionId() {
return versionId;
}
public void setVersionId(Long versionId) {
this.versionId = versionId;
}

public String getName() {
return name;
}
```

```
 public void setName(String name) {
 this.name = name;
 }

 public Long getCampignId() {
 return campignId;
}
private void setCampignId(Long campignId) {
 this.campignId = campignId;
}

 }
```

Step 2.
In the .hbm.xml file

```
<class name="beans.Campign" table="CAMPIGN" optimistic-lock="version">

      <id name="campignId" type="long" column="cid">
      <generator class="sequence">
            <param name="sequence">CAMPIGN_ID_SEQ</param>

      </generator>
   </id>
<version name="versionId" type="long" column="version" />

      <property name="name" column="c_name"/>

  </class>
```

Step 3.
Create a coulmn name "version" in the CAMPIGN table.

Step 4.
In the code

```
// foo is an instance loaded by a previous Session
session = sf.openSession();
int oldVersion = foo.getVersion();
session.load( foo, foo.getKey() );
if ( oldVersion!=foo.getVersion ) throw new StaleObjectStateException();
foo.setProperty("bar");
session.flush();
session.connection().commit();
```

session.close();

You can handle StaleObjectStateException() and do what ever you want.
You can display error message.

Hibernate autumatically create/update the version number when you update/insert any row in the table.

Question No : 9
How to handle user think time using hibernate ?
Answer :
version checking used in hibernate when more then one thread trying to access same data.
For example :
User A edit the row of the TABLE for update (In the User Interface changing data - This is user thinking time)
and in the same time User B edit the same record for update and click the update.
Then User A click the Update and update done. Chnage made by user B is gone.

In hibernate you can perevent slate object updatation using version checking.

Check the version of the row when you are upding the row.
Get the version of the row when you are fetching the row of the TABLE for update.
On the time of updation just fetch the version number and match with your version number (on the time of fetching).

This way you can prevent slate object updatation.

Steps 1:
Declare a variable "versionId" in your Class with setter and getter.

```
public class Campign {
private Long versionId;
private Long campignId;
private String name;
public Long getVersionId() {
return versionId;
}
public void setVersionId(Long versionId) {
this.versionId = versionId;
}

public String getName() {
```

```
 return name;
 }
 public void setName(String name) {
 this.name = name;
 }

 public Long getCampignId() {
 return campignId;
}
private void setCampignId(Long campignId) {
 this.campignId = campignId;
}

 }
```

Step 2.
In the .hbm.xml file

```
<class name="beans.Campign" table="CAMPIGN" optimistic-lock="version">

      <id name="campignId" type="long" column="cid">
      <generator class="sequence">
            <param name="sequence">CAMPIGN_ID_SEQ</param>

      </generator>
   </id>
<version name="versionId" type="long" column="version" />

      <property name="name" column="c_name"/>

   </class>
```

Step 3.
Create a coulmn name "version" in the CAMPIGN table.

Step 4.
In the code

```
// foo is an instance loaded by a previous Session
session = sf.openSession();
int oldVersion = foo.getVersion();
session.load( foo, foo.getKey() );
if ( oldVersion!=foo.getVersion ) throw new StaleObjectStateException();
foo.setProperty("bar");
```

```
session.flush();
session.connection().commit();
session.close();
```

You can handle StaleObjectStateException() and do what ever you want.
You can display error message.

Hibernate autumatically create/update the version number when you update/insert any row in the table.

Question No : 10
Q.Transaction with plain JDBC in Hibernate ?
Answer :
If you don't have JTA and don't want to deploy it along with your application, you will usually have to fall back to JDBC transaction demarcation. Instead of calling the JDBC API you better use Hibernate's Transaction and the built-in session-per-request functionality:

To enable the thread-bound strategy in your Hibernate configuration:

set hibernate.transaction.factory_class to org.hibernate.transaction.JDBCTransactionFactory
set hibernate.current_session_context_class to thread

```
Session session = factory.openSession();
Transaction tx = null;
try {
    tx = session.beginTransaction();

    // Do some work
    session.load(...);
    session.persist(...);

    tx.commit(); // Flush happens automatically
}
catch (RuntimeException e) {
    tx.rollback();
    throw e; // or display error message
}
finally {
    session.close();
}
```

Question No : 11

Q.What are the general considerations or best practices for defining your Hibernate persistent classes?

Answer :

1.You must have a default no-argument constructor for your persistent classes and there should be getXXX() (i.e accessor/getter) and setXXX(i.e. mutator/setter) methods for all your persistable instance variables.

2.You should implement the equals() and hashCode() methods based on your business key and it is important not to use the id field in your equals() and hashCode() definition if the id field is a surrogate key (i.e. Hibernate managed identifier). This is because the Hibernate only generates and sets the field when saving the object.

3. It is recommended to implement the Serializable interface. This is potentially useful if you want to migrate around a multi-processor cluster.

4.The persistent class should not be final because if it is final then lazy loading cannot be used by creating proxy objects.

Question No : 12

Q.Difference between session.update() and session.lock() in Hibernate ?

Answer :

Both of these methods and saveOrUpdate() method are intended for reattaching a detached object.

The session.lock() method simply reattaches the object to the session without checking or updating the database on the assumption that the database in sync with the detached object.

It is the best practice to use either session.update(..) or session.saveOrUpdate().

Use session.lock() only if you are absolutely sure that the
detached object is in sync with your detached object or if it does not matter because
you will be overwriting all the columns that would have changed later on within the same transaction.

Each interaction with the persistent store occurs in a new Session. However, the same persistent instances are reused for each interaction with the database. The application manipulates the state of detached instances originally loaded in another Session and then "reassociates" them using Session.update() or Session.saveOrUpdate().

// foo is an instance loaded by a previous Session

```
foo.setProperty("bar");
session = factory.openSession();
session.saveOrUpdate(foo);
session.flush();
session.connection().commit();
session.close();
```

You may also call lock() instead of update() and use LockMode.READ (performing a version check, bypassing all caches) if you are sure that the object has not been modified.

Question No : 13

Q.Difference between getCurrentSession() and openSession() in Hibernate ?

Answer :

getCurrentSession() :

The "current session" refers to a Hibernate Session bound by Hibernate behind the scenes, to the transaction scope.

A Session is opened when getCurrentSession() is called for the first time and closed when the transaction ends.

It is also flushed automatically before the transaction commits. You can call getCurrentSession() as often and anywhere you want as long as the transaction runs.

To enable this strategy in your Hibernate configuration:

set hibernate.transaction.manager_lookup_class to a lookup strategy for your JEE container
set hibernate.transaction.factory_class to org.hibernate.transaction.JTATransactionFactory

Only the Session that you obtained with sf.getCurrentSession() is flushed and closed automatically.

Example :

```
try {
    UserTransaction tx = (UserTransaction)new InitialContext()
                        .lookup("java:comp/UserTransaction");

    tx.begin();

    // Do some work
    sf.getCurrentSession().createQuery(...);
    sf.getCurrentSession().persist(...);

    tx.commit();
}
catch (RuntimeException e) {
    tx.rollback();
```

```
    throw e; // or display error message
}
```

openSession() :
If you decide to use manage the Session yourself the go for sf.openSession() , you have to flush() and close() it.
It does not flush and close() automatically.
Example :

```
UserTransaction tx = (UserTransaction)new InitialContext()
                        .lookup("java:comp/UserTransaction");

Session session = factory.openSession();

try {
    tx.begin();

    // Do some work
    session.createQuery(...);
    session.persist(...);

    session.flush(); // Extra work you need to do

    tx.commit();
}
catch (RuntimeException e) {
    tx.rollback();
    throw e; // or display error message
}
finally {
    session.close(); // Extra work you need to do
}
```

Question No : 14
Difference between session.saveOrUpdate() and session.merge()?
Answer :
saveOrUpdate() does the following:

? if the object is already persistent in this session, do nothing

? if another object associated with the session has the same identifier, throw an exception

? if the object has no identifier property, save() it

? if the object's identifier has the value assigned to a newly instantiated object, save() it

? if the object is versioned (by a <version> or <timestamp>), and the version property value is the same

value assigned to a newly instantiated object, save() it

? otherwise update() the object

merge() is very different:

? if there is a persistent instance with the same identifier currently associated with the session, copy the state

of the given object onto the persistent instance

? if there is no persistent instance currently associated with the session, try to load it from the database, or

create a new persistent instance

? the persistent instance is returned

? the given instance does not become associated with the session, it remains detached

---

Question No : 15

Filter in Hibernate with Example?

Answer :

Filter in Hibernate ------

USER (ID INT, USERNAME VARCHAR, ACTIVATED BOOLEAN) - TABLE

public class User

{

private int id;

```
private String username;

private boolean activated;




public boolean isActivated()

{

return activated;

}

public void setActivated(boolean activated)

{

this.activated = activated;

}

public int getId()

{

return id;

}

public void setId(int id)

{

this.id = id;

}

public String getUsername()
```

```
{
return username;
}
public void setUsername(String username)
{
this.username = username;
}
}
```

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-mapping
PUBLIC "-//Hibernate/Hibernate Mapping DTD//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
<class name="User">
<id name="id" type="int">
<generator class="native"/>
</id>

<property name="username" type="string" length="32"/>
```

<property name="activated" type="boolean"/>

<filter name="activatedFilter" condition=":activatedParam = activated"/>

</class>

<filter-def name="activatedFilter">

<filter-param name="activatedParam" type="boolean"/>

</filter-def>

</hibernate-mapping>

Save and Fetch using filter example

```
User user1 = new User();

user1.setUsername("name1");

user1.setActivated(false);


session.save(user1);


User user2 = new User();

user2.setUsername("name2");

user2.setActivated(true);
```

```
session.save(user2);

User user3 = new User();
user3.setUsername("name3");
user3.setActivated(true);

session.save(user3);

User user4 = new User();
user4.setUsername("name4");
user4.setActivated(false);

session.save(user4);
```

All the four user saved to Data Base User Table.

Now Fetch the User using Filter..

```
Filter filter = session.enableFilter("activatedFilter");
filter.setParameter("activatedParam",new Boolean(true));
Query query = session.createQuery("from User");
```

```
Iterator results = query.iterate();

while (results.hasNext())

{

User user = (User) results.next();

System.out.print(user.getUsername() + " is ");



}
```

Guess the Result :

name2 name3

Because Filer is filtering (only true value) data before query execute.

Question No : 16

Q.How does Value replacement in Message Resource Bundle work?

Answer :

In the resource bundle file, you can define a template like:

errors.required={0} is required.

ActionErrors errors = new ActionErrors();

errors.add(ActionErrors.GLOBAL_ERROR,

new ActionError("error.custform","First Name"));

Then the Error message is : First Name is required.

Other constructors are

public ActionError(String key, Object value0, Object value1)

. . .

public ActionError(String key, Object[] values);

Question No : 17

Difference between list() and iterate() i9n Hibernate?

Answer :

If instances are already be in the session or second-level cache iterate() will give better performance.

If they are not already cached, iterate() will be slower

than list() and might require many database hits for a simple query.

Question No : 18

Difference between session.load() and session.get() ?

Answer :

load() will throw an unrecoverable exception if there is no matching database row.

get() will return null if there is no matching database row.

Cat fritz = (Cat) session.load(Cat.class, "1");

Return the Cat Object with key 1. If there is no Cat Object with key 1 then throw will throw an unrecoverable exception.

If the class is

mapped with a proxy, load() just returns an uninitialized proxy and does not actually hit the database until you

invoke a method of the proxy. This behaviour is very useful if you wish to create an association to an object

without actually loading it from the database. It also allows multiple instances to be loaded as a batch if batchsize

is defined for the class mapping.

Cat fritz = (Cat) session.get(Cat.class, "1");

If you are not certain that a matching row exists, you should use the get() method, which hits the database immediately

and returns null if there is no matching row.

Question No : 19

Deleting persistent objects

Answer :

Session.delete() will remove an object's state from the database. Of course, your application might

still hold
a reference to a deleted object. It's best to think of delete() as making a persistent instance transient.
sess.delete(cat);

Question No : 20
SQL statements execution order.
Answer :
1. all entity insertions, in the same order the corresponding objects were saved using Session.save()
2. all entity updates
3. all collection deletions
4. all collection element deletions, updates and insertions
5. all collection insertions
6. all entity deletions, in the same order the corresponding objects were deleted using Session.delete()

Question No : 21
Difference between session.saveOrUpdate() and session.merge();
Answer :
saveOrUpdate() does the following:
? if the object is already persistent in this session, do nothing
? if another object associated with the session has the same identifier, throw an exception
? if the object has no identifier property, save() it
? if the object's identifier has the value assigned to a newly instantiated object, save() it
? if the object is versioned (by a <version> or <timestamp>), and the version property value is the same
value assigned to a newly instantiated object, save() it
? otherwise update() the object

merge() is very different:
? if there is a persistent instance with the same identifier currently associated with the session, copy the state
of the given object onto the persistent instance
? if there is no persistent instance currently associated with the session, try to load it from the database, or
create a new persistent instance
? the persistent instance is returned
? the given instance does not become associated with the session, it remains detached
---
Question No : 22
Modifying persistent objects?

Answer :
DomesticCat cat = (DomesticCat) sess.load(Cat.class, new Long(69));
cat.setName("PK");
sess.flush(); // changes to cat are automatically detected and persisted To Data Base.

No need any session.update() call.

Question No : 23
SQL Queries In Hibernate..
Answer :
You may express a query in SQL, using createSQLQuery() and let Hibernate take care of the mapping from
result sets to objects. Note that you may at any time call session.connection() and use the JDBC Connection
directly. If you chose to use the Hibernate API, you must enclose SQL aliases in braces:
List cats = session.createSQLQuery(
"SELECT {cat.*} FROM CAT {cat} WHERE ROWNUM<10",
"cat",
Cat.class
).list();
List cats = session.createSQLQuery(
"SELECT {cat}.ID AS {cat.id}, {cat}.SEX AS {cat.sex}, " +
"{cat}.MATE AS {cat.mate}, {cat}.SUBCLASS AS {cat.class}, ... " +
"FROM CAT {cat} WHERE ROWNUM<10",
"cat",
Cat.class
).list()
SQL queries may contain named and positional parameters, just like Hibernate queries.

Question No : 24
Filter in Hibernate
Answer :
Filter in Hibernate ------
USER (ID INT, USERNAME VARCHAR, ACTIVATED BOOLEAN) - TABLE

```
public class User
{
    private int id;
    private String username;
    private boolean activated;
```

```
    public boolean isActivated()
    {
        return activated;
    }
    public void setActivated(boolean activated)
    {
        this.activated = activated;
    }
    public int getId()
    {
        return id;
    }
    public void setId(int id)
    {
        this.id = id;
    }
    public String getUsername()
    {
        return username;
    }
    public void setUsername(String username)
    {
        this.username = username;
    }
}
-----------------------------------------------------------------
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-mapping
   PUBLIC "-//Hibernate/Hibernate Mapping DTD//EN"
   "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
  <class name="User">
    <id name="id" type="int">
      <generator class="native"/>
    </id>

    <property name="username" type="string" length="32"/>
    <property name="activated" type="boolean"/>
    <filter name="activatedFilter" condition=":activatedParam = activated"/>
```

```
  </class>
  <filter-def name="activatedFilter">
    <filter-param name="activatedParam" type="boolean"/>
  </filter-def>
</hibernate-mapping>
```

Save and Fetch using filter example

```
User user1 = new User();
user1.setUsername("name1");
user1.setActivated(false);

session.save(user1);

User user2 = new User();
user2.setUsername("name2");
user2.setActivated(true);

session.save(user2);

User user3 = new User();
user3.setUsername("name3");
user3.setActivated(true);

session.save(user3);

User user4 = new User();
user4.setUsername("name4");
user4.setActivated(false);

session.save(user4);
```

All the four user saved to Data Base User Table.

Now Fetch the User using Filter..

```
Filter filter = session.enableFilter("activatedFilter");
filter.setParameter("activatedParam",new Boolean(true));
Query query = session.createQuery("from User");
        Iterator results = query.iterate();
        while (results.hasNext())
```

```
{
    User user = (User) results.next();
    System.out.print(user.getUsername() + " is ");

}
```

Guess the Result :
name2
name3

Because Filer is filtering (only true value) data before query execute.

Question No : 25
Criteria Query Two Condition
Answer :
Criteria Query Two Condition- Example

```
<class name="com.bean.Organization" table="ORGANIZATION">
    <id name="orgId" column="ORG_ID" type="long">
      <generator class="native"/>
    </id>
<property name="organizationName" column="ORGANISATION_NAME" type="string"
length="500"/>
<property name="town" column="TOWN" type="string"      length="200"/>
<property name="statusCode" column="STATUS" type="string"      length="1"/>
</class>
```

List of organisation where town equals to pune and status = "A".

```
List organizationList = session.createCriteria(Organization.class)
.add(Restrictions.eq("town","pune"))
.add(Restrictions.eq("statusCode","A"))
.list();
```

Question No : 26
Equal and Not Equal criteria query.
Answer :
Equal and Not Equal criteria query- Example

```
<class name="com.bean.Organization" table="ORGANIZATION">
    <id name="orgId" column="ORG_ID" type="long">
      <generator class="native"/>
    </id>
<property name="organizationName" column="ORGANISATION_NAME" type="string"
```

length="500"/>
<property name="town" column="TOWN" type="string" length="200"/>
</class>
List of organisation where town equals to pune.

List organizationList =
session.createCriteria(Organization.class).add(Restrictions.eq("town","pune")).list();

List of organisation where town not equals pune.

List organizationList =
session.createCriteria(Organization.class).add(Restrictions.ne("town","pune")).list();

Question No : 27
Cascade Save or Update in Hibernate ?
Answer :
Cascade Save or Update - In one to Many- EXAMPLE
PROCESS_TYPE_LOV (PROCESS_TYPE_ID number, PROCESS_TYPE_NAME varchar) - TABLE
PROCESS (PROCESS_ID number,PROCESS_NAME varchar,PROCESS_TYPE_ID number)- TABLE

```
public class ProcessTypeBean {

 private Long processTypeId;
 private String processTypeName;


 /**
  * @return Returns the processTypeId.
  */
 public Long getProcessTypeId() {
  return processTypeId;
 }
 /**
  * @param processTypeId The processTypeId to set.
  */
 public void setProcessTypeId(Long processTypeId) {
  this.processTypeId = processTypeId;
 }
 /**
  * @return Returns the processTypeName.
```

```
 */
 public String getProcessTypeName() {
  return processTypeName;
 }
 /**
  * @param processTypeName The processTypeName to set.
  */
 public void setProcessTypeName(String processTypeName) {
  this.processTypeName = processTypeName;
 }

}

public class ProcessBean {

 private Long processId;
 private String processName = "";
 private ProcessTypeBean processType;

 public Long getProcessId() {
  return processId;
 }
 /**
  * @param processId The processId to set.
  */
 public void setProcessId(Long processId) {
  this.processId = processId;
 }
 /**
  * @return Returns the processName.
  */
 public String getProcessName() {
  return processName;
 }
 /**
  * @param processName The processName to set.
  */
 public void setProcessName(String processName) {
  this.processName = processName;
 }
 /**
  * @return Returns the processType.
```

```
  */
 public ProcessTypeBean getProcessType() {
  return processType;
 }
 /**
  * @param processType The processType to set.
  */
 public void setProcessType(ProcessTypeBean processType) {
  this.processType = processType;
 }
}

<class name="com.bean.ProcessBean"
  table="PROCESS">
  <id name="processId" type="long" column="PROCESS_ID" />
  <property name="processName" column="PROCESS_NAME" type="string"
   length="50" />
  <many-to-one name="processType" column="PROCESS_TYPE_ID" class="ProcessTypeBean"
cascade="save-update" />

</class>

<class name="com.bean.ProcessTypeBean"
  table="PROCESS_TYPE_LOV">
  <id name="processTypeId" type="long" column="PROCESS_TYPE_ID" />
  <property name="processTypeName" column="PROCESS_TYPE_NAME"
   type="string" length="50" />

 </class>
--------------------------------------------------------------------------------
Save Example Code -

ProcessTypeBean pstype = new ProcessTypeBean();
pstype.setProcessTypeName("Java Process");

        ProcessBean process = new ProcessBean();
        process.setProcessName("Production")
        ProcessBean.setProcessType(pstype);

//        session.save(pstype);  -- This save not required because of in the mapping file
cascade="save-update"
        session.save(process); - This will insert both ProcessBean and ProcessTypeBean;
```

Question No : 28
One To Many Bi-directional Relation in Hibernate?
Answer :
Bi-DireCtional One to Many Relation- EXAMPLE
PROCESS_TYPE_LOV (PROCESS_TYPE_ID number, PROCESS_TYPE_NAME varchar) - TABLE
PROCESS (PROCESS_ID number,PROCESS_NAME varchar,PROCESS_TYPE_ID number)- TABLE

```
public class ProcessTypeBean {

 private Long processTypeId;
 private String processTypeName;
        private List processes = null;

 /**
  * @return Returns the processes.
  */
 public List getProcesses() {
  return processes;
 }
 /**
  * @param processes The processes to set.
  */
 public void setProcesses(List processes) {
  this.processes = processes;
 }

 /**
  * @return Returns the processTypeId.
  */
 public Long getProcessTypeId() {
  return processTypeId;
 }
 /**
  * @param processTypeId The processTypeId to set.
  */
 public void setProcessTypeId(Long processTypeId) {
  this.processTypeId = processTypeId;
```

```
 }
 /**
  * @return Returns the processTypeName.
  */
 public String getProcessTypeName() {
  return processTypeName;
 }
 /**
  * @param processTypeName The processTypeName to set.
  */
 public void setProcessTypeName(String processTypeName) {
  this.processTypeName = processTypeName;
 }

}

public class ProcessBean {

 private Long processId;
 private String processName = "";
 private ProcessTypeBean processType;

 public Long getProcessId() {
  return processId;
 }
 /**
  * @param processId The processId to set.
  */
 public void setProcessId(Long processId) {
  this.processId = processId;
 }
 /**
  * @return Returns the processName.
  */
 public String getProcessName() {
  return processName;
 }
 /**
  * @param processName The processName to set.
  */
 public void setProcessName(String processName) {
  this.processName = processName;
```

```
 }
 /**
  * @return Returns the processType.
  */
 public ProcessTypeBean getProcessType() {
  return processType;
 }
 /**
  * @param processType The processType to set.
  */
 public void setProcessType(ProcessTypeBean processType) {
  this.processType = processType;
 }
}

<class name="com.bean.ProcessBean"
  table="PROCESS">
  <id name="processId" type="long" column="PROCESS_ID" />
  <property name="processName" column="PROCESS_NAME" type="string"
   length="50" />
  <many-to-one name="processType" column="PROCESS_TYPE_ID" lazy="false" />

</class>

<class name="com.bean.ProcessTypeBean"
  table="PROCESS_TYPE_LOV">
  <id name="processTypeId" type="long" column="PROCESS_TYPE_ID" />
  <property name="processTypeName" column="PROCESS_TYPE_NAME"
   type="string" length="50" />

  <bag name="processes" inverse="true" cascade="delete" lazy="false">
   <key column="PROCESS_TYPE_ID" />
   <one-to-many
    class="com.bean.ProcessBean" />
  </bag>

 </class>
```

Question No : 29

One To Many Mapping Using List ?

Answer :

WRITER (ID INT,NAME VARCHAR) - TABLE

STORY (ID INT,INFO VARCHAR,PARENT_ID INT) - TABLE
One writer can have multiple stories..

Mapping File...

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping
    PUBLIC "-//Hibernate/Hibernate Mapping DTD//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd">

<hibernate-mapping>
    <class name="Writer" table="WRITER">
         <id name="id" unsaved-value="0">
            <generator class="increment"/>
        </id>

        <list name="stories" cascade="all">
            <key column="parent_id"/>
             <one-to-many class="Story"/>
        </list>
        <property name="name" type="string"/>
    </class>
    <class name="Story"
         table="story">
         <id name="id" unsaved-value="0">
            <generator class="increment"/>
        </id>
          <property name="info"/>
    </class>
</hibernate-mapping>
```

```
public class Writer {
  private int id;
  private String name;
  private List stories;


  public void setId(int i) {
    id = i;
  }

  public int getId() {
    return id;
```

```
  }

  public void setName(String n) {
    name = n;
  }

  public String getName() {
    return name;
  }

  public void setStories(List l) {
    stories = l;
  }

  public List getStories() {
    return stories;
  }
}

---------------------------------------------------
public class Story {
  private int id;
  private String info;

  public Story(){
  }

  public Story(String info) {
    this.info = info;
  }

  public void setId(int i) {
    id = i;
  }

  public int getId() {
    return id;
  }

  public void setInfo(String n) {
    info = n;
  }
```

```
  public String getInfo() {
    return info;
  }
}
```

Save Example ..

```
Writer wr = new Writer();
wr.setName("Das");

      ArrayList list = new ArrayList();
      list.add(new Story("Story Name 1"));
      list.add(new Story("Story Name 2"));
      wr.setStories(list);

      Transaction transaction = null;

      try {
          transaction = session.beginTransaction();
          session.save(sp);
          transaction.commit();
      } catch (Exception e) {
          if (transaction != null) {
            transaction.rollback();
            throw e;
          }
      }  finally {
          session.close();
      }
```

Question No : 30

Many To Many Relation In Hibernate ?

Answer :

Best Example..for Many to Many in Hibernate ..

EVENTS (uid int, name VARCHAR) Table

SPEAKERS (uid int, firstName VARCHAR) Table

EVENT_SPEAKERS (elt int, event_id int, speaker_id int) Table

```
import java.util.Set;
import java.util.HashSet;
```

```
public class Speaker{

    private Long id;
    private String firstName;
    private Set events;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

   public Set getEvents() {
        return this.events;
    }

    public void setEvents(Set events) {
        this.events = events;
    }

    private void addEvent(Event event) {
        if (events == null) {
            events = new HashSet();
        }
        events.add(event);
    }
}
```

```
import java.util.Date;
import java.util.Set;

public class Event{
```

```
  private Long id;
  private String name;
  private Set speakers;

  public void setId(Long id) {
    this.id = id;
  }

  public Long getId() {
    return id;
  }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setSpeakers(Set speakers) {
    this.speakers = speakers;
  }

  public Set getSpeakers() {
    return speakers;
  }

 }
```

Event.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 2.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd">

<hibernate-mapping>
    <class name="Event" table="events">
        <id name="id" column="uid" type="long" unsaved-value="null">
            <generator class="increment"/>
        </id>
```

```
        <property name="name" type="string" length="100"/>
       <set name="speakers" table="event_speakers" cascade="all">
            <key column="event_id"/>
            <many-to-many class="Speaker"/>
        </set>
    </class>
</hibernate-mapping>
```

Speaker.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 2.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd">

<hibernate-mapping>
    <class name="Speaker" table="speakers">
        <id name="id" column="uid" type="long">
            <generator class="increment"/>
        </id>
        <property name="firstName" type="string" length="20"/>
        <set name="events" table="event_speakers" cascade="all">
            <key column="speaker_id"/>
            <many-to-many class="Event"/>
        </set>
    </class>
</hibernate-mapping>
```

Save and Fetch Example

```
Event event = new Event();
event.setName("Inverse test");
event.setSpeakers(new HashSet());
event.getSpeakers().add(new Speaker("Ram", event));
event.getSpeakers().add(new SpeakerManyToMany("Syam", event));
event.getSpeakers().add(new SpeakerManyToMany("Jadu",  event));
session.save(event); /// Save All the Data

event = (Event) session.load(Event.class, event.getId());
Set speakers = event.getSpeakers();

        for (Iterator i = speakers.iterator(); i.hasNext();) {
            Speaker speaker = (Speaker) i.next();
```

```
        System.out.println(speaker.getFirstName());
        System.out.println(speaker.getId());
    }
```

Question No : 31

What does session.refresh() do ?

Answer :

It is possible to re-load an object and all its collections at any time, using the refresh() method. This is useful
when database triggers are used to initialize some of the properties of the object.
For Example - Triger on cat_name coulmn. Trigger is updating hit_count coulmn in the same Cat Table. When Insert data into Cat TABLE trigger update hit_count coulmn to 1. sess.refresh() reload all the data. No nned again to select call.

```
sess.save(cat);
sess.flush(); //force the SQL INSERT
sess.refresh(cat); //re-read the state (after the trigger executes)
```

Question No : 32

Difference between session.load() and session.get()?

Answer :

```
Cat fritz = (Cat) session.load(Cat.class, "1");
```

Return the Cat Object with key 1. If there is no Cat Object with key 1 then throw will throw an unrecoverable exception.

If the class is
mapped with a proxy, load() just returns an uninitialized proxy and does not actually hit the database until you
invoke a method of the proxy. This behaviour is very useful if you wish to create an association to an object
without actually loading it from the database. It also allows multiple instances to be loaded as a batch if batchsize
is defined for the class mapping.

```
Cat fritz = (Cat) session.get(Cat.class, "1");
```

If you are not certain that a matching row exists, you should use the get() method, which hits the database immediately
and returns null if there is no matching row.

Question No : 33
Hibernate setup using .cfg.xml file ?
Answer :
The XML configuration file is by default expected to be in the root o your CLASSPATH. Here is an example:

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<!-- a SessionFactory instance listed as /jndi/name -->
<session-factory
name="java:hibernate/SessionFactory">
<!-- properties -->
<property name="connection.datasource">java:/comp/env/jdbc/MyDB</property>
<property name="dialect">org.hibernate.dialect.MySQLDialect</property>
<property name="show_sql">false</property>
<property name="transaction.factory_class">
org.hibernate.transaction.JTATransactionFactory
</property>
<property name="jta.UserTransaction">java:comp/UserTransaction</property>
<!-- mapping files -->
<mapping resource="org/hibernate/auction/Cost.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

As you can see, the advantage of this approach is the externalization of the mapping file names to configuration.
The hibernate.cfg.xml is also more convenient once you have to tune the Hibernate cache. Note that is
your choice to use either hibernate.properties or hibernate.cfg.xml, both are equivalent, except for the
above mentioned benefits of using the XML syntax.
With the XML configuration, starting Hibernate is then as simple as

```
SessionFactory sf = new Configuration().configure().buildSessionFactory();
```

You can pick a different XML configuration file using

```
SessionFactory sf = new Configuration()
.configure("catdb.cfg.xml")
.buildSessionFactory();
```

Cost.hbm.xml -----> looks like

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
      "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
      "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

 <hibernate-mapping>
 <class name="com.bean.Cost" table="COST">

      <id name="id"  column="ID">
      </id>
      <property name="isQueued" type="int" column="IS_QUEUED"/>
      <property name="queueDate" type="timestamp" column="QUEUE_DATE"/>
      <property name="lastModifiedDate" type="timestamp" column="LAST_MODIFIED_DATE"/>
      <property name="lastModifiedBy" column="LAST_MODIFIED_BY"/>
      <property name="amount" column="AMOUNT" type="double"/>
      <property name="currencyCode" column="CURRENCY_CODE" />
      <property name="year" column="YEAR"/>
      <property name="quarter" column="QUARTER"/>
      <property name="costModFlag" type="int" column="COST_MOD_FLAG"/>
      <property name="parentId" column="PARENT_ID"/>
      <property name="oldParentId" column="OLD_PARENT_ID"/>
      <property name="parentIdModFlag" type="int" column="PARENT_ID_MOD_FLAG"/>
      <property name="dateIncurred" type="timestamp" column="DATE_INCURRED"/>
      <property name="USDAmount" column="USD_AMOUNT" type="double"/>
      <property name="isDeleted" type="int" column="IS_DELETED"/>
      </class>

 </hibernate-mapping>
```

Question No : 34

How to add .hbm.xml file in sessionFactory?

Answer :

```
SessionFactory sf = new Configuration()
.addFile("Item.hbm.xml")
.addFile("Bid.hbm.xml")
.buildSessionFactory();
```

Question No : 35

How to get Hibernate statistics ?

Answer :

SessionFactory.getStatistics() is give you all the statistics .

Question No : 36

How to set 2nd level cache in hibernate with EHCache?

Answer :

When you are creating SessionFactory just add the below steps

```
String ecache = appHome+File.separatorChar+"ehcache.xml";
try {
            CacheManager.create(ecache);
        } catch (CacheException e) {
            // logger.logError(e);
        }*/
```

Then

```
sessionFactory = configuration.buildSessionFactory();
```

ECache.xml is like

```
<ehcache>
 <diskStore path="java.io.tmpdir"/>
<defaultCache
        maxElementsInMemory="10000"
        eternal="false"
        timeToIdleSeconds="120"
        timeToLiveSeconds="120"
        overflowToDisk="true"
        diskPersistent="false"
        diskExpiryThreadIntervalSeconds="120"
        />

    <cache name="bean.ApplicationBean"
        maxElementsInMemory="300"
        eternal="false"
        overflowToDisk="false"
        />
</ehcache>
```

ApplicationBean will be avilable in 2nd level cache

Question No : 37
How to get JDBC connections in hibernate?
Answer :
User Session.connection() method to get JDBC Connection.

Question No : 38
How will you configure Hibernate?
Answer :
Step 1> Put Hibernate properties in the classpath.
Step 2> Put .hbm.xml in class path ?
Code is Here to create session ...

```
package com.dao;




import java.io.File;
import java.io.FileInputStream;

import java.util.Properties;




import org.apache.log4j.Logger;
import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

/**
 *
 * @author Satya Das
 */
public class HibernateUtil {
 protected static final Logger logger=Logger.getLogger(HibernateUtil.class);
 public static String  appHome = "No";
```

```
private static SessionFactory sessionFactory;


private static final ThreadLocal threadSession = new ThreadLocal();
private static final ThreadLocal threadTransaction = new ThreadLocal();



/**
 *  Initialize Hibernate Configuration
 */
public static void initMonitor(){
 logger.info("Hibernate configure");
 try {
  logger.info("appHome"+appHome);
  String path_properties = appHome+File.separatorChar+"hibernate.properties";
  String path_mapping = appHome+File.separatorChar+"mapping_classes.mysql.hbm.xml";
  //String ecache = appHome+File.separatorChar+"ehcache.xml";


  Properties propHibernate = new Properties();
  propHibernate.load(new FileInputStream(path_properties));

  Configuration configuration = new Configuration();
  configuration.addFile(path_mapping);
  configuration.setProperties(propHibernate);

  /* try {
             CacheManager.create(ecache);
         } catch (CacheException e) {
             // logger.logError(e);
         }*/

  sessionFactory = configuration.buildSessionFactory();



 } catch (Throwable ex) {
  logger.error("Exception in initMonitor",ex);
  throw new ExceptionInInitializerError(ex);
```

```
 }
}
/**
 * @return a Session Factory Object
 */
public static SessionFactory getSessionFactory() {
 logger.info("Inside getSessionFactory method");
 try {

  if (sessionFactory == null) {
   initMonitor();
  }else {

   //sessionFactory.getStatistics().logSummary();
  }


 } catch (Exception e) {
  logger.error("Exception in getSessionFactory",e);
 }

 return sessionFactory;
}

/**
 * @return  Session . Start a Session
 */
public static Session getSession() {

 Session s = (Session) threadSession.get();
 logger.debug("session"+s);
 if (s == null) {

  s = getSessionFactory().openSession();
  threadSession.set(s);
  logger.debug("session 1 $"+s);
 }
 return s;
}

/**
 *  Close Session
```

```
 */
public  static void closeSession(){

 Session s = (Session) threadSession.get();
 threadSession.set(null);
 if (s != null && s.isOpen()) {
  s.flush();
  s.close();
 }
}


/**
 * Start a new database transaction.
 */
public static void beginTransaction(){
 Transaction tx = null;

 if (tx == null) {
  tx = getSession().beginTransaction();
  threadTransaction.set(tx);
 }
}

/**
 * Commit the database transaction.
 */
public static void commitTransaction(){
 Transaction tx = (Transaction) threadTransaction.get();
 try {
  if ( tx != null ) {
   tx.commit();
  }

  threadTransaction.set(null);

 } catch (HibernateException ex) {
  rollbackTransaction();

  throw ex;
 }
}
```

```
 /**
  * Rollback the database transaction.
  */
 public static void rollbackTransaction(){

  Transaction tx = (Transaction) threadTransaction.get();
  try {
   threadTransaction.set(null);
   if ( tx != null && !tx.wasCommitted() && !tx.wasRolledBack() ) {
    tx.rollback();
   }
  } finally {
   closeSession();
  }
 }




}
```

Question No : 39

How to create Session and SessionFactory in Hibernate?

Answer :

Step 1> Put Hibernate properties in the classpath.

Step 2> Put .hbm.xml in class path ?

Code is Here to create session ...

```
package com.dao;




import java.io.File;
```

```
import java.io.FileInputStream;

import java.util.Properties;




import org.apache.log4j.Logger;
import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

/**
 *
 * @author Satya Das
 */
public class HibernateUtil {
 protected static final Logger logger=Logger.getLogger(HibernateUtil.class);
 public static String  appHome = "No";


 private static SessionFactory sessionFactory;


 private static final ThreadLocal threadSession = new ThreadLocal();
 private static final ThreadLocal threadTransaction = new ThreadLocal();



 /**
  *  Initialize Hibernate Configuration
  */
 public static void initMonitor(){
  logger.info("Hibernate configure");
  try {
   logger.info("appHome"+appHome);
   String path_properties = appHome+File.separatorChar+"hibernate.properties";
   String path_mapping = appHome+File.separatorChar+"mapping_classes.mysql.hbm.xml";
```

```
  //String ecache = appHome+File.separatorChar+"ehcache.xml";


  Properties propHibernate = new Properties();
  propHibernate.load(new FileInputStream(path_properties));

  Configuration configuration = new Configuration();
  configuration.addFile(path_mapping);
  configuration.setProperties(propHibernate);

  /* try {
              CacheManager.create(ecache);
          } catch (CacheException e) {
              // logger.logError(e);
          }*/

  sessionFactory = configuration.buildSessionFactory();



 } catch (Throwable ex) {
  logger.error("Exception in initMonitor",ex);
  throw new ExceptionInInitializerError(ex);
 }
}
/**
 * @return a Session Factory Object
 */
public static SessionFactory getSessionFactory() {
 logger.info("Inside getSessionFactory method");
 try {

  if (sessionFactory == null) {
   initMonitor();
  }else {

   //sessionFactory.getStatistics().logSummary();
  }


 } catch (Exception e) {
  logger.error("Exception in getSessionFactory",e);
```

```
 }

 return sessionFactory;
}

/**
 * @return  Session . Start a Session
 */
public static Session getSession() {

 Session s = (Session) threadSession.get();
 logger.debug("session"+s);
 if (s == null) {

  s = getSessionFactory().openSession();
  threadSession.set(s);
  logger.debug("session 1 $"+s);
 }
 return s;
}

/**
 *  Close Session
 */
public  static void closeSession(){

 Session s = (Session) threadSession.get();
 threadSession.set(null);
 if (s != null && s.isOpen()) {
  s.flush();
  s.close();
 }
}



/**
 * Start a new database transaction.
 */
public static void beginTransaction(){
 Transaction tx = null;

 if (tx == null) {
```

```
  tx = getSession().beginTransaction();
  threadTransaction.set(tx);
 }
}

/**
 * Commit the database transaction.
 */
public static void commitTransaction(){
 Transaction tx = (Transaction) threadTransaction.get();
 try {
  if ( tx != null ) {
   tx.commit();
  }

  threadTransaction.set(null);

 } catch (HibernateException ex) {
  rollbackTransaction();

  throw ex;
 }
}



/**
 * Rollback the database transaction.
 */
public static void rollbackTransaction(){

 Transaction tx = (Transaction) threadTransaction.get();
 try {
  threadTransaction.set(null);
  if ( tx != null && !tx.wasCommitted() && !tx.wasRolledBack() ) {
   tx.rollback();
  }
 } finally {
  closeSession();
 }
}
```

}

Question No : 40
What are the Instance states in Hibernate?
Answer :
transient
The instance is not, and has never been associated with any persistence context. It has no persistent identity
(primary key value).
persistent
The instance is currently associated with a persistence context. It has a persistent identity (primary key
value) and, perhaps, a corresponding row in the database. For a particular persistence context, Hibernate
guarantees that persistent identity is equivalent to Java identity (in-memory location of the object).
detached
The instance was once associated with a persistence context, but that context was closed, or the instance
was serialized to another process. It has a persistent identity and, perhaps, a corrsponding row in the database.
For detached instances, Hibernate makes no guarantees about the relationship between persistent
identity and Java identity.

Question No : 41
What are the core components in Hibernate ?
Answer :
SessionFactory (org.hibernate.SessionFactory)
A threadsafe (immutable) cache of compiled mappings for a single database. A factory for Session and a
client of ConnectionProvider. Might hold an optional (second-level) cache of data that is reusable between transactions, at a process- or cluster-level.
Session (org.hibernate.Session)
A single-threaded, short-lived object representing a conversation between the application and the persistent
store. Wraps a JDBC connection. Factory for Transaction. Holds a mandatory (first-level) cache of persistent
objects, used when navigating the object graph or looking up objects by identifier.
Persistent objects and collections

Short-lived, single threaded objects containing persistent state and business function. These might be ordinary
JavaBeans/POJOs, the only special thing about them is that they are currently associated with (exactly
one) Session. As soon as the Session is closed, they will be detached and free to use in any application
layer (e.g. directly as data transfer objects to and from presentation).
Transient and detached objects and collections
Instances of persistent classes that are not currently associated with a Session. They may have been instantiated
by the application and not (yet) persisted or they may have been instantiated by a closed Session.
Transaction (org.hibernate.Transaction)
(Optional) A single-threaded, short-lived object used by the application to specify atomic units of work.
Abstracts application from underlying JDBC, JTA or CORBA transaction. A Session might span several
Transactions in some cases. However, transaction demarcation, either using the underlying API or Transaction,
is never optional!
Architecture
Hibernate 3.0.2 9
ConnectionProvider (org.hibernate.connection.ConnectionProvider)
(Optional) A factory for (and pool of) JDBC connections. Abstracts application from underlying Datasource
or DriverManager. Not exposed to application, but can be extended/implemented by the developer.
TransactionFactory (org.hibernate.TransactionFactory)
(Optional) A factory for Transaction instances. Not exposed to the application, but can be extended/
implemented by the developer.
Extension Interfaces
Hibernate offers many optional extension interfaces you can implement to customize the behavior of your
persistence layer. See the API documentation for details.

Question No : 42
What is a Hibernate Session? Can you share a session object between different theads?

Answer :
Session is a light weight and a non-threadsafe object (No, you cannot share it between threads) that represents a single unit-of-work with the database. Sessions are opened by a SessionFactory and then are closed when all work is complete. Session is the primary interface for the persistence

service. A session obtains a database connection lazily (i.e. only when required). To avoid creating too many sessions ThreadLocal class can be used as shown below to get the current session no matter how many times you make call to the currentSession() method.

```
?
public class HibernateUtil {
?
public static final ThreadLocal local = new ThreadLocal();

public static Session currentSession() throws HibernateException {
Session session = (Session) local.get();
//open a new session if this thread has no session
if(session == null) {
session = sessionFactory.openSession();
local.set(session);
}
return session;
}
}
```

Question No : 43

addScalar() method in hibernate...

Answer :

```
Double max = (Double) sess.createSQLQuery("select max(cat.weight) as maxWeight from cats cat")
.addScalar("maxWeight", Hibernate.DOUBLE);
.uniqueResult();
```

addScalar() method confim that maxWeight is always double type.

This way you don't need to check for it is double or not.

Question No : 44

Hibernate session.close does _not_ call session.flush ?

Answer :

session.close() don't call session.flush() before closing the session.

This is the session.close() code in hibernate.jar

```
public Connection close() throws HibernateException {
  log.trace( "closing session" );
  if ( isClosed() ) {
   throw new SessionException( "Session was already closed" );
  }


  if ( factory.getStatistics().isStatisticsEnabled() ) {
   factory.getStatisticsImplementor().closeSession();
  }

  try {
   try {
    if ( childSessionsByEntityMode != null ) {
     Iterator childSessions = childSessionsByEntityMode.values().iterator();
     while ( childSessions.hasNext() ) {
      final SessionImpl child = ( SessionImpl ) childSessions.next();
      child.close();
     }
    }
   }
   catch( Throwable t ) {
    // just ignore
   }

   if ( rootSession == null ) {
    return jdbcContext.getConnectionManager().close();
   }
   else {
    return null;
   }
  }
  finally {
   setClosed();
   cleanup();
  }
 }
```

Question No : 45

What is the main difference between Entity Beans and Hibernate ?

Answer :

1)In Entity Bean at a time we can interact with only one data Base. Where as in Hibernate we can

able to establishes the connections to more than One Data Base. Only thing we need to write one more configuration file.

2) EJB need container like Weblogic, WebSphare but hibernate don't nned. It can be run on tomcat.

3) Entity Beans does not support OOPS concepts where as Hibernate does.

4) Hibernate supports multi level cacheing, where as Entity Beans doesn't.

5) In Hibernate C3P0 can be used as a connection pool.

6) Hibernate is container independent. EJB not.

Question No : 46
Difference between session.save() and session.saveOrUpdate()?
Answer :
session.save() - Insert data into the table
session.saveOrUpdate()- Insert data if primary key not exist otherwise update

---

Question No : 47
How are joins handled using Hinernate.
Answer :

Best is use Criteria query
Example -
You have parent class

```
public class Organization {
private long orgId;
private List messages;
}
```

Child class

```
public class Message {
 private long messageId;
private Organization organization;
}
```

.hbm.xml file

```
  <class name="com.bean.Organization" table="ORGANIZATION">
<bag name="messages" inverse="true" cascade="delete" lazy="false">
   <key column="MSG_ID" />
   <one-to-many
    class="com.bean.Message" />
  </bag>



    </class>

<class name="com.bean.Message" table="MESSAGE">
 <many-to-one name="organization" column="ORG_ID" lazy="false"/>

    </class>
```

Get all the messages from message table where organisation id = <any id>

Criteria query is :

```
session.createCriteria(Message.class).createAlias("organization","org").
   add(Restrictions.eq("org.orgId",new Long(orgId))).add(Restrictions.in("statusCode",status)).list();
```

you can get all the details in hibernate website.

http://www.hibernate.org/hib_docs/reference/en/html/associations.html

---

Question No : 48

What is Hibernate proxy?

Answer :

By default Hibernate creates a proxy for each of the class you map in mapping file. This class contain the code to invoke JDBC. This class is created by hibernate using CGLIB.

Proxies are created dynamically by subclassing your object at runtime. The subclass has all the methods of the parent, and when any of the methods are accessed, the proxy loads up the real object from the DB and calls the method for you. Very nice in simple cases with no object hierarchy. Typecasting and instanceof work perfectly on the proxy in this case since it is a direct subclass.

Question No : 49

What is the main advantage of using the hibernate than using the sql ?
Answer :
1) If you are using Hibernate then you don't need to learn specific SQL (like oracle,mysql), You have to user POJO class object as a table.

2) Don't need to learn query tuning..Hibernate criteria query automatically tuned the query for best performance.

3) You can use inbuild cache for storing data

4) No need to create own connection pool , we can use c3po . It will give best result...

5> Don't need any join query which reduce performance and complex. Using hibernate you have to define in bean and hbm.xml file.

6> You can add filter in Hibernate which exceute before you query fires and get the best performance

7> EhCache is used for 2nd level cache to store all the redefind data like country table ..

Question No : 50
how to create primary key using hibernate?
Answer :
```
<id name="userId" column="USER_ID" type="int">
        <generator class="increment"/>
    </id>
```

increment generator class automatically generate the primary key for you.

Question No : 51
what is the advantage of Hibernate over jdbc?
Answer :
There are lots
1) If you are using Hibernate then you don't need to learn specific SQL (like oracle,mysql), You have to user POJO class object as a table.
2) Don't need to learn query tuning..Hibernate criteria query automatically tuned the query for best performance.
3) You can use inbuild cache for storing data
4) No need to create own connection pool , we can use c3po . It will give best result...

Question No : 52

How to Execute Stored procedure in Hibernate ?
Answer :
Option 1:

```
Connection con = null;



        try {
            con = session.connection();

            CallableStatement st = con
                    .prepareCall("{call your_sp(?,?)}");
            st.registerOutParameter(2, Types.INTEGER);
            st.setString(1, "some_Seq");

            st.executeUpdate();
```

Option 2:

```
<sql-query name="selectAllEmployees_SP" callable="true">
<return alias="emp" class="employee">
   <return-property name="empid" column="EMP_ID"/>

  <return-property name="name" column="EMP_NAME"/>
  <return-property name="address" column="EMP_ADDRESS"/>
    { ? = call selectAllEmployees() }
</return>
</sql-query>
```

code :

```
SQLQuery sq = (SQLQuery) session.getNamedQuery("selectAllEmployees_SP");

List results = sq.list();
```

Question No : 53
what is lazy fetching in hibernate?
Answer :
Lazy setting decides whether to load child objects while loading the Parent Object.You need to do this setting respective hibernate mapping file of the parent class.Lazy = true (means not to load child)By default the lazy loading of the child objects is true. This make sure that the child objects are not loaded unless they are explicitly invoked in the application by calling getChild() method on parent.In this case hibernate issues a fresh database call to load the child when getChild() is

actully called on the Parent object.But in some cases you do need to load the child objects when parent is loaded. Just make the lazy=false and hibernate will load the child when parent is loaded from the database.Exampleslazy=true (default)Address child of User class can be made lazy if it is not required frequently.lazy=falseBut you may need to load the Author object for Book parent whenever you deal with the book for online bookshop

---

****

****

Java Interview Questions

****

Question No : 1

How to retrieve a key and Value from a map ?

Answer :

get the code example

[URL=http://www.techforum360.com/forum/viewthread?thread=145]http://www.techforum360.com/forum/viewthread?thread=145[/URL]

Question No : 2

Why can an interface supertype call methods that belong to Object class?

Answer :

The members of an interface are:

1) Those members declared in the interface.

2) Those members inherited from direct superinterfaces.

3) If an interface has no direct superinterfaces, then the interface implicitly declares a public abstract member method m with signature s, return type r, and throws clause t corresponding to each public instance method m with signature s, return type r, and throws clause t declared in Object, unless a method with the same signature, same return type, and a compatible throws clause is explicitly declared by the interface. It is a compile-time error if the interface explicitly declares such a method m in the case where m is declared to be final in Object.

Question No : 3

Q. What is difference between shallow copy and deep copy ?

Answer :

Only instances of classes that implement the Cloneable interface can be cloned. Trying to clone an object that does not implement the Cloneable interface throws a CloneNotSupportedException. Both shallow copy and deep copy object nedd to implement Cloneable Interface.

Shallow copy :

The java.lang.Object root superclass defines a clone() method.
default behavior of clone() is to return a shallow copy of the object. This means that the values of all of the original object?s fields are copied to the fields of the new object.
If a shallow copy is performed on obj1, then it is copied but its contained objects are not.
For Example:
Public class Emp
{
Private Address address;
}
Emp emp1 = new Emp();
Address add = new Address();
emp1. address= add.

If we clone
Emp emp2 = emp1.clone();
Then emp2.addess reference to the same Address object which emp1 refer.

Deep Copy :

A deep copy occurs when an object is copied along with the objects to which it refers.
A deep copy makes a distinct copy of each of the object?s fields, recursing through the entire graph of other objects referenced by the object being copied. The Java API provides no deep-copy equivalent to Object.clone(). One solution is to simply implement your own custom method (e.g., deepCopy()) that returns a deep copy of an instance of one of your classes.
Your Object Class should implements Serializable and Cloneable interface. .
public static Object copy(Object orig) {

```
    Object obj = null;

    try {

        // Write the object out to a byte array

        ByteArrayOutputStream bo = new ByteArrayOutputStream();

        ObjectOutputStream out = new ObjectOutputStream(bo);

        out.writeObject(orig);

        out.flush();
```

```
        out.close();

         // Make an input stream from the byte array and read

        // a copy of the object back in.

        ObjectInputStream in = new ObjectInputStream(

            new ByteArrayInputStream(bos.toByteArray()));

        obj = in.readObject();

    }

    catch(IOException e) {

        e.printStackTrace();

    }

    catch(ClassNotFoundException cnfe) {

        cnfe.printStackTrace();

    }

    return obj;

  }
```

Question No : 4

Q.Why to override equals() and hashCode()? and How i can implement both equals() and hashCode() for Set ?

Answer :

If you are implementing HashSet to store unique object then you need to implement equals() and hashcode() method.

if two objects are equal according to the equals() method, they must have the same hashCode() value (although the reverse is not generally true).

Two scenarios

Case 1) : If you don't implement equals() and hashcode() method :

When you are adding objects to HashSet , HashSet checks for uniqueness using equals() and hashcode() method the class (ex. Emp class). If there is no equals() and hashcode() method the Emp class then HashSet checks default Object classes equals() and hashcode() method.

In the Object class , equals method is

```
public boolean equals(Object obj) {
 return (this == obj);
   }
```

Under this default implementation, two references are equal only if they refer to the exact same object. Similarly, the default implementation of hashCode() provided by Object is derived by mapping the memory address of the object to an integer value.

This will fail to check if two Emp object with same employee name .

For Example :

```
Emp emp1 = new Emp();
 emp1.setName("sat");
Emp emp2 = new Emp();
  emp2.setName("sat");
```

Both the objects are same but based on the default above method both objects are dirrefent because references and hashcode are different .

So in the HashSet you can find the duplicate emp objects.

To overcome the issue equals() and hashcode() method need to override.

Case 2) : If you override equals() and hashcode() method

Example : implement equals and hashcode

```
public class Emp {
 private long empId;
 String name = "";
public boolean equals(Object o) {
  if (o == this)
  return true;
  if (!(o instanceof Emp))
  return false;
  Emp emp = (Emp)o;
  return emp. empId == empId &&
  emp. name == name;
```

```
  }

 public int hashCode(){
  int result = 10;
  result = result * new Integer(String.valueOf(empId)).intValue();
  return result;
 }
}
```

In the equals() , it check for name is same or not. This way you can find out the objects are equals or not.

In the hashCode() also it return some unique value for each object. In this way if two Emp object has same empId then it will say both are same object.

Now HashSet store only unique objects.

If you do

```
Emp emp1 = new Emp();
 emp1.setName("sat");
Emp emp2 = new Emp();
  emp2.setName("sat");

if(emp1.equals(emp2)){
   System.out.println("equal");
  }
```

This will print : equal

Question No : 5

Q.How to sort list of objects (User Defined) using comparator? in Descending Order.

Answer :

Q. How to sort list of objects (User Defined) using comparator?

You can use Collections.sort(List,Comparator) to sort objects.

Example : You have User Bean , you want to sort based on username filed.

User Class :

```
   public class User {
   String userName = "";
   String city = "";
   String state = "";
 /**  * @return Returns the userName.
  */
 public String getUserName() {
  return userName;
 }
 /**
```

```
  * @param userName The userName to set.
  */
 public void setUserName(String userName) {
  this.userName = userName;
 }
}
```

Then you have to write a comparator class.

```
public class UserNameComparator implements Comparator {
   public int compare(Object user, Object anotherUser) {

     String firstName1 = ((User) user).getUserName().toUpperCase();

     String firstName2 = ((User) anotherUser).getUserName().toUpperCase();

       return firstName1.compareTo(firstName2);

   }
 }
```

Now sorting code

```
                    User user1 = new User();
  user1.setUserName("das");
  User user2 = new User();
  user2.setUserName("nick");
  User user3 = new User();
  user3.setUserName("ram");
  User user4 = new User();
  user4.setUserName("jadu");

  ArrayList list = new ArrayList();
  list.add(user1);
  list.add(user2);
  list.add(user3);
  list.add(user4);

  Collections.sort(list,new UserNameComparator()); // sort ascending order.

// Descending order
Collections.reverse(list);

  for(int i=0;i<list.size();i++){
```

```
  User usr = (User)list.get(i);
  System.out.println(usr.getUserName());
 }
```

Question No : 6

Q. How to sort list of objects (User Defined) using comparator?

Answer :

You can use Collections.sort(List,Comparator) to sort objects.

Example : You have User Bean , you want to sort based on username filed.

User Class :

```
  public class User {
  String userName = "";
  String city = "";
  String state = "";
 /**  * @return Returns the userName.
  */
 public String getUserName() {
  return userName;
 }
 /**
  * @param userName The userName to set.
  */
 public void setUserName(String userName) {
  this.userName = userName;
 }
}
```

Then you have to write a comparator class.

```
public class UserNameComparator implements Comparator {
  public int compare(Object user, Object anotherUser) {

    String firstName1 = ((User) user).getUserName().toUpperCase();

    String firstName2 = ((User) anotherUser).getUserName().toUpperCase();

      return firstName1.compareTo(firstName2);

  }
 }
```

Now sorting code

```
                User user1 = new User();
  user1.setUserName("das");
  User user2 = new User();
  user2.setUserName("nick");
  User user3 = new User();
  user3.setUserName("ram");
  User user4 = new User();
  user4.setUserName("jadu");

  ArrayList list = new ArrayList();
  list.add(user1);
  list.add(user2);
  list.add(user3);
  list.add(user4);

  Collections.sort(list,new UserNameComparator()); // sort ascending order.

  for(int i=0;i<list.size();i++){
   User usr = (User)list.get(i);
   System.out.println(usr.getUserName());
  }
```

Question No : 7

What is the difference between int and Interger?

Answer :

int is primitive type and Integer is Wrapper Class.

For Example :

```
int i=2;
Integer i = new Integer (2);
```

You can have question like why we need Wrapper Class(Integer)?

Answer is , if you are using Collection objects like Hashtable etc. you can't use int as a key .you have to use object so you can use Integer class.

```
Hashtable ht = new Hashtable ();
ht.put(2, "USA") ;// You can not do this

ht.put(new Integer(2), "USA") ;// you have to do like this.
```

Question No : 8
Singleton Double-checked locking in Java?

Answer :

```
public static Singleton getInstance()
{
  if (instance == null)
  {
    synchronized(Singleton.class) {  //1
      if (instance == null)          //2
        instance = new Singleton();  //3
    }
  }
  return instance;
}
```

Thread 1 enters the getInstance() method.

Thread 1 enters the synchronized block at //1 because instance is null.

Thread 1 is preempted by thread 2.

Thread 2 enters the getInstance() method.

Thread 2 attempts to acquire the lock at //1 because instance is still null. However, because thread 1 holds the lock, thread 2 blocks at //1.

Thread 2 is preempted by thread 1.

Thread 1 executes and because instance is still null at //2, creates a Singleton object and assigns its reference to instance.

Thread 1 exits the synchronized block and returns instance from the getInstance() method.

Thread 1 is preempted by thread 2.

Thread 2 acquires the lock at //1 and checks to see if instance is null.

Because instance is non-null, a second Singleton object is not created and the one created by thread 1 is returned.

Question No : 9

What are the parameters to follow Creating and Destroying Objects in Java?

Answer :

Item 1: Consider providing static factory methods instead of
constructors

```
public static Boolean valueOf(boolean b) {
return (b ? Boolean.TRUE : Boolean.FALSE);
}
```

advantage of static factory methods is that, unlike constructors, they are not required to create a new object each time they're invoked.

This allows immutable classes
(Item 13) to use preconstructed instances or to cache instances as they're constructed and to dispense these instances repeatedly so as to avoid creating unnecessary duplicate objects.

The Boolean.valueOf(boolean) method illustrates this technique: It never creates an object.

This technique can greatly improve performance if equivalent objects are requested frequently, especially if these objects are expensive to create.

it allows an immutable class to ensure that no two equal instances exist:
a.equals(b) if and only if a==b. If a class makes this guarantee, then its clients can use the == operator instead of the equals(Object) method, which may result in a substantial performance improvement

implements this
optimization, and the String.intern method implements it in a limited form.

advantage of static factory methods is that, unlike constructors, they can return an object of any subtype of their return type. This gives you great flexibility in choosing the class of the returned object.

One application of this flexibility is that an API can return objects without making their classes public. Hiding implementation classes in this fashion can lead to a very compact API.

Item 2: Enforce the singleton property with a private constructor

Item 4: Avoid creating duplicate objects

It is often appropriate to reuse a single object instead of creating a new functionally equivalent object each time it is needed. Reuse can be both faster and more stylish. An object can always be reused if it is immutable

As an extreme example of what not to do, consider this statement:

String s = new String("silly"); // DON'T DO THIS!

The statement creates a new String instance each time it is executed, and none of those object creations is necessary. The argument to the String constructor ("silly") is itself a String instance, functionally identical to all of the objects created by the constructor. If this usage occurs in a loop or in a frequently invoked method, millions of String instances can be created needlessly.

The improved version is simply the following:

String s = "No longer silly";

Item 5: Eliminate obsolete object references

So where is the memory leak? If a stack grows and then shrinks, the objects that were popped off the stack will not be garbage collected, even if the program using the stack has no more references to them. This is because the stack maintains obsolete references to these objects. An obsolete reference is simply a reference that will never be dereferenced again. In this case, any references outside of the ?active portion? of the element array are obsolete. The active portion consists of the elements whose index is less than size.

```
public Object pop() {
if (size == 0)
throw new EmptyStackException();
return elements[--size];
}
```

The fix for this sort of problem is simple: Merely null out references once they become obsolete. In the case of our Stack class, the reference to an item becomes obsolete as soon as it's popped off the stack. The corrected version of the pop method looks like this:

```
public Object pop() {
if (size==0)
throw new EmptyStackException();
Object result = elements[--size];
elements[size] = null; // Eliminate obsolete reference
return result;
}
```

Item 6: Avoid finalizers

There is no guarantee that finalizers will be executed promptly. It can take arbitrarily long between the time that an object becomes unreachable and the time that its finalizer is executed. This means that nothing time-critical should ever be done by a

finalizer. For example, it is a grave error to depend on a finalizer to close open files because open file descriptors are a limited resource. If many files are left open because the JVM is tardy in executing finalizers, a program may fail because it can no longer open files.

Question No : 10

Q.What are the different scopes for Java variables?

Answer :

The scope of a Java variable is determined by the context in which the variable is declared. Thus a java variable can have one of the three scopes at any given point in time.

1. Instance : - These are typical object level variables, they are initialized to default values at the time of creation of object, and remain accessible as long as the object accessible.
2. Local : - These are the variables that are defined within a method. They remain accessbile only during the course of method excecution. When the method finishes execution, these variables fall out of scope.
3. Static: - These are the class level variables. They are initialized when the class is loaded in JVM for the first time and remain there as long as the class remains loaded. They are not tied to any particular object instance

---

Question No : 11

Q.What method must be implemented by all threads?

Answer :

All tasks must implement the run() method, whether they are a subclass of Thread or implement the Runnable interface.

Question No : 12

Q.Can an unreachable object become reachable again?

Answer :

An unreachable object may become reachable again. This can happen when the object's finalize() method is invoked and the object performs an operation which causes it to become accessible to reachable objects

---

Question No : 13

Q.What are the steps in the JDBC connection?

Answer :
While making a JDBC connection we go through the following steps :

Step 1 : Register the database driver by using :
Class.forName(\" driver classs for that specific database\");

Class.forName("com.mysql.jdbc.Driver");

Step 2 : Now create a database connection using :

Connection con = DriverManager.getConnection(url,username,password);

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/art",art,art);

Step 3: Now Create a query using :

Statement stmt = Connection.Statement(\"select * from EMP\");

Step 4 : Exceute the query :

ResultSet rs = stmt.exceuteQuery();

```
while(rs.next()){
System.out.println(rs.getString(1));
}
```

Question No : 14
Q.How are this() and super() used with constructors?

Answer :
This() is used to invoke a constructor of the same class. super() is used to invoke a superclass constructor
---
Question No : 15
Q.What is the difference between static and non-static variables?

Answer :
A static variable is associated with the class as a whole rather than with specific instances of a class. Non-static variables take on unique values with each object instance.
For Example :

```
public class Test{
 static int i=5;
 int j=7;
}
Test t1= new Test ();
Test t2= new Test ();
t1.j=10;
t2.j=15;
```

but for the case of static i ,

```
t1.i=10;
t2.i=15;
```

System.out.println(t2.i) will give you 15 not 10 . because i is sharable and related to class not instance.

Question No : 16
Q.What is the purpose of finalization?

Answer :
The purpose of finalization is to give an unreachable object the opportunity to perform any cleanup processing before the object is garbage collected.

Question No : 17
Q.Does garbage collection guarantee that a program will not run out of memory?

Answer :
Garbage collection does not guarantee that a program will not run out of memory. It is possible for programs to use up memory resources faster than they are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection

---

Question No : 18
Q.What is synchronization and why is it important?

Answer :
With respect to multithreading, synchronization is the capability to control

the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to significant errors

---

Question No : 19

Q.What is daemon thread and which method is used to create the daemon thread?

Answer :

Daemon thread is a low priority thread which runs intermittently in the back ground doing the garbage collection operation for the java runtime system. setDaemon method is used to create a daemon thread.

Question No : 20

Q.What are synchronized methods and synchronized statements?

Answer :

Synchronized methods are methods that are used to control access to an object. A thread only executes a synchronized method after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement

---

Question No : 21

Q.If I write System.exit (0); at the end of the try block, will the finally block still execute?

Answer :

No ,in this case the finally block will not execute because when you say System.exit (0); the control immediately goes out of the program, and thus finally never executes.

Question No : 22

Q.If I write return at the end of the try block, will the finally block still execute?

Answer :

Yes even if you write return as the last statement in the try block and no exception occurs, the finally block will execute. The finally block will execute and then the control return

---

Question No : 23

Q.Is it necessary that each try block must be followed by a catch block?

Answer :

It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method

---

Question No : 24

Q.What is the basic difference between the 2 approaches to exception handling.
1> try catch block and
2> specifying the candidate exceptions in the throws clause?
When should you use which approach?

Answer :

In the first approach as a programmer of the method, you urself are dealing with the exception. This is fine if you are in a best position to decide should be done in case of an exception. Whereas if it is not the responsibility of the method to deal with it's own exceptions, then do not use this approach. In this case use the second approach.
In the second approach we are forcing the caller of the method to catch the exceptions, that the method is likely to throw. This is often the approach library creators use. They list the exception in the throws clause and we must catch them.

Question No : 25

Q.What are the different ways to handle exceptions?

Answer :

There are two ways to handle exceptions,
1. By wrapping the desired code in a try block followed by a catch block to catch the exceptions. and
2. List the desired exceptions in the throws clause of the method and let the caller of the method hadle those exceptions.

Question No : 26

Q.What is the difference between error and an exception?

Answer :

An error is an irrecoverable condition occurring at runtime. Such as OutOfMemory error. These JVM errors and you can not repair them at runtime. While exceptions are conditions that occur because of bad input etc. e.g. FileNotFoundException will be thrown if the specified file does not exist. Or a NullPointerException will take place if you try using a null reference. In most of the cases it is possible to recover from an exception (probably by giving user a feedback for entering proper values etc.).

Question No : 27

Q: What is wrapper class? Explain with example?

Answer :
Java provides specialized classes corresponding to each of the primitive data types. These are called wrapper classes.
They are e.g. Integer, Character, Double etc.
All wrapper classes are final. You can't extend the class.

Question No : 28
Q.What is serialization? Explain with example?
Answer :
Serialization involves saving the current state of an object to a stream, and restoring an equivalent object from that stream.

For an object to be serialized, it must be an instance of a class that implements either the Serializable or Externalizable interface. Both interfaces only permit the saving of data associated with an object's variables.
They depend on the class definition being available to the Java Virtual Machine at reconstruction time in order to construct the object.
The Serializable interface relies on the Java runtime default mechanism to save an object's state.
Writing an object is done via the writeObject() method in the ObjectOutputStream class (or the ObjectOutput interface).
Writing a primitive value may be done through the appropriate write<datatype>() method.
Reading the serialized object is accomplished using the readObject() method of the ObjectInputStream class, and primitives may be read using the various read<datatype>() methods.

The Externalizable interface specifies that the implementing class will handle the serialization on its own, instead of relying on the default runtime mechanism.
This includes which fields get written (and read), and in what order.
The class must define a writeExternal() method to write out the stream, and a corresponding readExternal() method to read the stream.
Inside of these methods the class calls ObjectOutputStream writeObject(), ObjectInputStream readObject(), and any necessary write<datatype>() and read<datatype>() methods, for the desired fields.

```
class DataOrder implements Serializable{
  String apples, peaches, pears, cardnum, custID;
  double icost;
  int itotal;
}
```

Saving the current state of an object to a stream
DataOrder orders = new DataOrder();

```
FileOutputStream fos =
      new FileOutputStream(orders);
    ObjectOutputStream oos =
      new ObjectOutputStream(fos);
    oos.writeObject(order);

restoring an equivalent object from that stream
FileInputStream fis =
          new FileInputStream(orders);
        ObjectInputStream ois =
          new ObjectInputStream(fis);
        order = (DataOrder)ois.readObject();
```

Question No : 29

Q.What one should take care of while serializing the object?

Answer :

make sure that all the included objects are also serializable. If any of the objects is not serializable then it throws a NotSerializableException.

Question No : 30

Q. When you serialize an object, what happens to the object references included in the object?

Answer :

The serialization mechanism generates an object graph for serialization. Thus it determines whether the included object references are serializable or not. This is a recursive process. Thus when an object is serialized, all the included objects are also serialized alongwith the original obect.

Question No : 31

Q.What happens to the static fields of a class during serialization?

Answer :

There are three exceptions in which serialization doesnot necessarily read and write to the stream. These are

1. Serialization ignores static fields, because they are not part of ay particular state state.
2. Base class fields are only hendled if the base class itself is serializable.
3. Transient fields.

Question No : 32
Q. Objects are passed by value or by reference?

Answer :
Java only supports pass by value. With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object.

Question No : 33
Q. Primitive data types are passed by reference or pass by value?

Answer :
Primitive data types are passed by value.

Question No : 34
Q. What type of parameter passing does Java support?

Answer :
Java only supports pass by value. With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object.

Question No : 35
Q.Can a top level class be private or protected?

Answer :
No. top level class can be public, abstract and final.

Question No : 36
Q.What is the default value of an object reference declared as an instance variable?

Answer :
null unless we define it explicitly.

Question No : 37
Q.What is the difference between declaring a variable and defining a variable?

Answer :
In declaration we just mention the type of the variable and it's name. We do not initialize it. But defining means declaration + initialization.
e.g String s; is just a declaration while String s = new String ("abcd"); Or String s = "abcd"; are both definitions.

Question No : 38
Q. What are different types of inner classes?

Answer :
Nested top-level classes, Member classes, Local classes, Anonymous classes

Nested top-level classes- If you declare a class within a class and specify the static modifier, the compiler treats the class just like any other top-level class.
Any class outside the declaring class accesses the nested class with the declaring class name acting similarly to a package. eg, outer.inner. Top-level inner classes implicitly have access only to static variables.There can also be inner interfaces. All of these are of the nested top-level variety.

Member classes - Member inner classes are just like other member methods and member variables and access to the member class is restricted, just like methods and variables. This means a public member class acts similarly to a nested top-level class. The primary difference between member classes and nested top-level classes is that member classes have access to the specific instance of the enclosing class.

Local classes - Local classes are like local variables, specific to a block of code. Their visibility is only within the block of their declaration. In order for the class to be useful beyond the declaration block, it would need to implement a
more publicly available interface.Because local classes are not members, the modifiers public, protected, private, and static are not usable.

Anonymous classes - Anonymous inner classes extend local inner classes one level further. As anonymous classes have no name, you cannot provide a constructor.

Question No : 39
Q. What is Overriding?
Answer :
When a class defines a method using the same name, return type, and arguments as a method in its superclass, the method in the class overrides the method in the superclass.
When the method is invoked for an object of the class, it is the new definition of the method that is called, and not the method definition from superclass.
Rules to folow :
Methods may be overridden to be more public, not more private.
Methods may be overridden to be thows the same exception or subclass of the exception thrown by the method of superclass.

Question No : 40
Q.What are Checked and UnChecked Exception?

Answer :
A checked exception is some subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses.
Making an exception checked forces client programmers to deal with the possibility that the exception will be thrown. eg, IOException thrown by java.io.FileInputStream's read() method· Unchecked exceptions are RuntimeException and any of its subclasses. Class Error and its subclasses also are unchecked. With an unchecked exception, however, the compiler doesn't force client programmers either to catch the
exception or declare it in a throws clause. In fact, client programmers may not even know that the exception could be thrown. eg, StringIndexOutOfBoundsException thrown by String's charAt() method· Checked exceptions must be caught at compile time. Runtime exceptions do not need to be. Errors often cannot be
---
Question No : 41
Q. What is final?

Answer :
A final class can't be extended ie., final class may not be subclassed. A final method can't be overridden when its class is inherited. You can't change value of a final variable (is a constant).

Question No : 42
Q.What is static in java?
Answer :
Static means one per class, not one for each object no matter how many instance of a class might exist.
This means that you can use them without creating an instance of a class.
Static methods are implicitly final, because overriding is done based on the type of the object, and static methods are attached to a class, not an object.
A static method in a superclass can be shadowed by another static method in a subclass, as long as the original method was not declared final.
However, you can't override a static method with a nonstatic method.
In other words, you can't change a static method into an instance method in a subclass.
Static varibales are not serialized.

Question No : 43
What is an abstract class?

Answer :
Abstract class must be extended/subclassed (to be useful). It serves as a template. A class that is abstract may not be instantiated (ie, you may not call its constructor), abstract class may contain static data. Any class with an abstract method is automatically abstract itself, and must be

declared as such.
A class may be declared abstract even if it has no abstract methods. This prevents it from being instantiated
---
Question No : 44
Q. What are the modifiers in Java ?
Answer :
public : Public class is visible in other packages, field is visible everywhere (class must be public too)
private : Private variables or methods may be used only by an instance of the same class that declares the variable or method, A private feature may only be accessed by the class that owns the feature.
protected : Is available to all classes in the same package and also available to all subclasses of the class that owns the protected feature.This access is provided even to subclasses that reside in a different package from the class that owns the protected feature.
default :What you get by default ie, without any access modifier (ie, public private or protected).It means that it is visible to all within a particular package
---
Question No : 45
Q.Difference between HashMap and HashTable?
Answer :
a)The key difference between the two is that access to the Hashtable is synchronized while access to the HashMap is not synchronized.
b)HashMap permits null values as Key, while Hashtable doesn't.
c)iterator in a HashMap is fail-safe(Fail-safe is relevant from the context of iterators.
If an iterator has been created on a collection object and some other thread tries to modify the collection object "structurally",
a concurrent modification exception will be thrown) while the enumerator of HashTable is not.

Question No : 46
Q.Difference between Vector and ArrayList?
Answer :
Vectors are synchronized. Any method that touches the Vector's contents is thread safe.
ArrayList, on the other hand, is unsynchronized, making them, therefore, not thread safe.
With that difference in mind, using synchronization will incur a performance hit.
So if you don't need a thread-safe collection, use the ArrayList don't use Vector.

A Vector defaults to doubling the size of its array, while the ArrayList increases its array size by 50 percent.
Depending on how you use these classes, you could end up taking a large performance hit while adding new elements.
If you don't know how much data you'll have, but you do know the rate at which it grows, Vector

does possess a slight advantage since you can set the increment value.

Both the ArrayList and Vector are good for retrieving elements from a specific position in the container or for adding and removing elements from the end of the container.
All of these operations can be performed in constant time -- O(1).
However, adding and removing elements from any other position proves more expensive -- linear to be exact: O(n-i), where n is the number of elements and i is the index of the element added or removed.
For Example : ArrayList Has 5 elements and you want to add an element on 2nd position.
Then arrayList add the element on 2nd position and shifted existing 2nd position element to 3rd , 3rd position element to 4th etc..
For the case of remove just opposite.
These operations are more expensive because you have to shift all elements at index i and higher over by one element

---

Question No : 47
Q.Difference between ArrayList and LinkedList?
Answer :
ArrayList is good for adding and removing elements from the end of the container. All of these operations can be performed in constant time -- O(1).
adding and removing elements from any other position proves more expensive -- linear to be exact: O(n-i), where n is the number of elements and i is the index of the element added or removed.
For Example : ArrayList Has 5 elements and you want to add an element on 2nd position.
Then arrayList add the element on 2nd position and shifted existing 2nd position element to 3rd , 3rd position element to 4th etc..

LinkedList can add or remove an element at any position in constant time -- O(1). Indexing an element is a bit slower -- O(i) where i is the index of the element in case of LinkedList.

Traversing an ArrayList is also easier since you can simply use an index instead of having to create an iterator.
The LinkedList creates an internal object for each element inserted. So you have to be aware of the extra garbage being created.

An ArrayList is a List implementation backed by a Java array.
With a LinkedList, the List implementation is backed by a doubly linked list data structure

---

Question No : 48
Q.What are pass by reference and passby value in Java?
Answer :
Pass By Reference means the passing the address itself rather than passing the value.

Passby Value means passing a copy of the value to be passed.
Java only supports pass by value. With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object.

Question No : 49
Q. When we go for Abstract and Interface in Java?
Answer :
When the sub-types behaviour is totally different then you use an interface, when the sub-types behaviour is partially common and different with respect to the supertype an abstract class is used.

In an abstract class the partially common behaviour is given a concrete implementation.

Since there is no common behaviour between an interface and a sub-type an interface does not have an implementation for any of its behaviour.

For Example - Abstract Class

```
abstract public  class Animal {
public eatChicken(){
System.out.println("Eat Chicken");
}

 }

 public Lion extends Animal{
 }
 public Tiger extends Animal{
 }
```

Both Lion and Tiger class has common behaviour eatChicken() and all the implementaion put into eatChicken() method of super calss.

so we go for abstract class.

Example- Interface

```
 Interface Animal{
 public eat();
 }
public Lion implements Animal{
 public eat(){
System.out.println("Lion eat Non Veg");
}
 }
public Cow implements Animal{
 public eat(){
```

```
  System.out.println("Cow eat Veg");
  }
   }
```

Both Lion and Cow class has different behaviour for eat() and they have different implementations. so we go for Interface.

Question No : 50

Q.What is the difference between interface and abstract class?

Answer :

* interface contains methods that must be abstract; abstract class may contain concrete methods.
* interface contains variables that must be static and final; abstract class may contain non-final and final variables.
* members in an interface are public by default, abstract class may contain non-public members.
* interface is used to "implements"; whereas abstract class is used to "extends".
* interface can be used to achieve multiple inheritance; abstract class can be used as a single inheritance.
* interface can "extends" another interface, abstract class can "extends" another class and "implements" multiple interfaces.
* interface is absolutely abstract; abstract class can be invoked if a main() exists.
* interface is more flexible than abstract class because one class can only "extends" one super class, but "implements" multiple interfaces.
* If given a choice, use interface instead of abstract class.

Question No : 51

Q.What are the Garbage collection algorithms in Java?

Answer :

Any garbage collection algorithm must do two basic things.

First, it must detect garbage objects.

Second, it must reclaim the heap space used by the garbage objects and make it available to the program.

Garbage detection is ordinarily accomplished by defining a set of roots and determining reachability from the roots.

An object is reachable if there is some path of references from the roots by which the executing program can access the object.

The roots are always accessible to the program. Any objects that are reachable from the roots are considered live.

Objects that are not reachable are considered garbage, because they can no longer affect the future course of program execution.

a. Reference counting collectors

Reference counting was an early garbage collection strategy.

here a reference count is maintained for each object.
When an object is first created its reference count is set to one.
When any other object or root is assigned a reference to that object, the object's count is incremented.
When a reference to an object goes out of scope or is assigned a new value, the object's count is decremented.
Any object with a reference count of zero can be garbage collected.
When an object is garbage collected, any objects that it refers to has their reference counts decremented.
In this way the garbage collection of one object may lead to the subsequent garbage collection of other objects.

b. Tracing collectors

Tracing garbage collectors trace out the graph of object references starting with the root nodes.
Objects that are encountered during the trace are marked in some way.
Marking is generally done by either setting flags in the objects themselves or by setting flags in a separate bitmap.
After the trace is complete, unmarked objects are known to be unreachable and can be garbage collected.
The basic tracing algorithm is called mark and sweep.
This name refers to the two phases of the garbage collection process.
In the mark phase, the garbage collector traverses the tree of references and marks each object it encounters.
In the sweep phase unmarked objects are freed, and the resulting memory is made available to the executing program.
In the JVM the sweep phase must include finalization of objects.

c. Compacting collectors

Garbage collectors of JVMs will likely have a strategy to combat heap fragmentation.
Two strategies commonly used by mark and sweep collectors are compacting and copying.
Both of these approaches move objects on the fly to reduce heap fragmentation.
Compacting collectors slide live objects over free memory space toward one end of the heap.
In the process the other end of the heap becomes one large contiguous free area.
All references to the moved objects are updated to refer to the new location.

d. Copying collectors

Copying garbage collectors move all live objects to a new area.
As the objects are moved to the new area, they are placed side by side,
thus eliminating any free spaces that may have separated them in the old area.
The old area is then known to be all free space.
The advantage of this approach is that objects can be copied as they are discovered by the traversal from the root nodes.

There are no separate mark and sweep phases.
Objects are copied to the new area on the fly, and forwarding pointers are left in their old locations.

The forwarding pointers allow objects encountered later in the traversal that refer to already copied objects to know the new location of the copied objects.

Question No : 52
Q.What is garbage collection and the purpose of garbage collection in Java?
Answer :
The JVM's heap stores all objects created by an executing Java program.
Objects are created by Java's "new" operator, and memory for new objects is allocated on the heap at run time.
Garbage collection is the process of automatically freeing objects that are no longer referenced by the program.
This frees the programmer from having to keep track of when to free allocated memory, thereby preventing many potential bugs and headaches.
The name "garbage collection" implies that objects that are no longer needed by the program are "garbage" and can be thrown away.
Other word we can say memory recycling.
When an object is no longer referenced by the program, the heap space it occupies must be recycled so that the space is available for subsequent new objects.
The garbage collector must somehow determine which objects are no longer referenced by the program and make available the heap space occupied by such unreferenced objects.
In the process of freeing unreferenced objects, the garbage collector must run any finalizers of objects being freed.

Question No : 53
Q. What is the difference between an Interface and an Abstract class?

Answer :
An abstract class can have instance methods that implement a default behavior. An Interface can only declare constants and instance methods, but cannot implement default behavior and all methods are implicitly abstract. An interface has all public members and no implementation. An abstract class is a class which may have the usual flavors of class members (private, protected, etc.), but has some abstract methods.

****

****

JSP Interview Questions
****

Question No : 1
How do i include static files within a jsp page ?
Answer :
This is static include..
you can just include using below code
<%@ include file="header.jsp" %>
or
<%@ include file="footer.html" %>

Dynamic include you can do like below
<jsp:include page="header.jsp"/>

or you can pass parameter also
<jsp:include page="header.jsp">
<jsp:param name="title" value="Java Interview Questions"/>
</jsp:include>

Question No : 2
Image upload from JSP to DataBase ?
Answer :
Image upload from JSP to Data Base step by step :

step 1 :
In the JSP
<form name="regform2" method="post" enctype="multipart/form-data">
<input type="file" name="ImageFile" id="ImageFile" onChange="uploadImage()"/>
</form>

java script :
function uploadImage(){
document.regform.action ="<%=request.getContextPath()%>/dfdmin?cmd=uploadimage";
document.regform.submit();
}

Step 2.
In the servlet - add the below code.

```
String rtempfile = File.createTempFile("temp","1").getParent();
MultipartRequest multi = new MultipartRequest(request, rtempfile, 500000 * 1024);
File rnewfile=null;
rnewfile = new
File(CommonArt.IMAGE_PATH+"jsp"+File.separator+"images"+File.separator+"uploadImage"+Fil
e.separator);
if(rnewfile.exists()){
}else{
rnewfile.mkdirs();
}

File f = multi.getFile("ImageFile");
System.out.println(f.getName());
FileInputStream fin =new FileInputStream(f);
RandomAccessFile r = new RandomAccessFile(rnewfile+File.separator+f.getName(),"rw");
filename = f.getName();
// FileOutputStream fos =new FileOutputStream(rnewfile);
byte sizefile[] = new byte[5000000];
fin.read(sizefile);
// fos.write(sizefile);
r.write(sizefile);
//fos.close();
r.close();
fin.close();
```

Step 3.
Insert into Database

```
InputStream is = new FileInputStream(f);
String sql = " INSERT INTO image_upload (IMAGE) VALUES (?) ";
pStmt = conn.prepareStatement(sql);
pStmt.setBinaryStream(1, is, (int)(f.length()));
pStmt.execute();
conn.commit();
```

This will upload multipart file to your data base.

Note : get cos.jar from oreilly website

Question No : 3
How to Disable session in JSP page ?
Answer :
Disabling the session in some pages will improve the performance of your JSP container.

Every time a JSP is requested, JSP creates an HttpSession object to maintain state for each unique client. The session data is accessible in the JSP as the implicit session object. In JSPs, sessions are enabled by default.
By default <%@ page session="true" %>

Session object uses the server resources. Each session object uses up a small amount of system resources as it is stored on the server side. This also increases the traffic as the session ID is sent from server to client. Client also sends the same session ID along with each request. If some of the JSP pages on your web site are getting millions of hits from internet browser and there is not need to identify the user, so its better to disable the session in that JSP page.

You can tell the container to disable session in the JSP file by setting the session attribute to false. Set the session attribute of the page directive to false.

<%@ page session="false" %>

```
<%@ page language="java" session="false"%>
<html>
<head>
<title>Session Disabled Example</title>
</head>
<body>
<p>Session is Disabled in this page </p>
</body>
</html>
```

Question No : 4
Q.How you do Session Management in JSP ?
Answer :
Http protocol is a stateless protocol, that means that it can't persist the data. Http treats each request as a new request so every time you will send a request you will be considered as a new user.

In session management whenever a request comes for any resource, a unique token is generated by the server and transmitted to the client by the response object and stored on the client machine as a cookie. We can also say that the process of managing the state of a web based client is through the use of session IDs. Session IDs are used to uniquely identify a client browser, while the server side processes are used to associate the session ID with a level of access. Thus, once a client has successfully authenticated to the web applicatiion, the session ID can be used as a stored authentication voucher so that the client does not have to retype their login information with each page request. Now whenever a request goes from this client again the ID or token will also be passed through the request object so that the server can understand from where the request is coming.
Session management can be achieved by :

1. Cookies: cookies are small bits of textual information that a web server sends to a browser and that browsers returns the cookie when it visits the same site again. In cookie the information is stored in the form of a name, value pair. By default the cookie is generated. If the user doesn't want to use cookies then it can disable them browser setting.

2. URL rewriting: In URL rewriting we append some extra information on the end of each URL that identifies the session. This URL rewriting can be used where a cookie is disabled. It is a good practice to use URL rewriting. In this session ID information is embedded in the URL, which is recieved by the application through Http GET requests when the client clicks on the links embedded with a page.
Exmaple : http://www.techfaq360.com/answers.jsp?sname=test

3. Hidden form fields: In hidden form fields the html entry will be like this : <input type ="hidden" name ="name" value="">. This means that when you submit the form, the specified name and value will be get included in get or post method. In this session ID information would be embedded within the form as a hidden field and submitted with the Http POST method.

4. HttpSession object :
   javax.servlet.http.HttpSession is an interface that provides a way to identify a user across more than one page request or visit to a web site. This is the way mainly used in webapplication. HttpSession object maintain session for you. You don't need to do any session management.

session.setAttribute("name",name);
String name = session.getAttribute("name");
you will get the same value which you have set.

Question No : 5
How can get the checked values from multiple check box checked ? or How to Handle multiple check box checked ?
Answer :
In the JSP page

```
for(int j = 0; j < messageList.size(); j++)    {

Message msg = (Message)messageList.get(j);

<input type="checkbox" name="delmesg_<%=msg.getMessageId()%>" value="on">

}
```

Retrive the check box which is checked if more then one

In the Servlet

```
Enumeration params = request.getParameterNames();

        List msgIdList = new ArrayList();

         while ( params != null && params.hasMoreElements()){

String param = (String)params.nextElement();

//System.out.println("param--------"+param);

if(params != null && param.startsWith("delmesg_")){


String msgId = param.substring(param.indexOf("_") + 1 ,param.length());

System.out.println("msgId"+msgId);

msgIdList.add(msgId);
```

```
}

    }



// msgIdList contains number of check box checled in the JSP
```

Question No : 6

Q.How to send email through JSP/Java?

Answer :

In the JSP

```
<%
            String from,

            String[] to,

            String subject,

            String textBody

            ) {

        boolean sentResult = false;



        try {

            Properties mailProps = new Properties();

            mailProps.put("mail.smtp.host", Common.SMTP_SERVER);

            Session session = Session.getDefaultInstance(mailProps, null);



            //create message

            MimeMessage message = new MimeMessage(session);
```

```
//set from
InternetAddress fromAdd = new InternetAddress(from);
message.setFrom(fromAdd);

//set to
InternetAddress[] toAdd = null;

if(to != null){
    toAdd = new InternetAddress[to.length];
    for(int i = 0 ; i < to.length; i++)
        toAdd[i] = new InternetAddress(to[i]);

    message.setRecipients(Message.RecipientType.TO,toAdd);
}

// set Subject
message.setSubject(subject);
```

```
    Multipart mp = new MimeMultipart("alternative");
    if(textBody != null){
  MimeBodyPart tbp = new MimeBodyPart();
  tbp.setText(textBody, "ISO-8859-1");
  tbp.setHeader("Content-Type", "text/plain; charset=" + "ISO-8859-1");
  mp.addBodyPart(tbp);
  }

    message.setContent(mp);

    //set sent Date
    message.setSentDate(new java.util.Date());

    // send the mail off

    Transport.send(message);
    sentResult = true;
 } catch(Exception exp) {
    exp.printStackTrace();
```

```
            sentResult = false;



        }

 %>
```

Question No : 7

Q.How to create and retrive a multiple selections list in jsp/servlet?

Answer :

This is the code to display multiple selections list in jsp.

List medList = DAO.getallMedium(); // this method retun list of Medium objects.

medList list contains list of Medium objects.

Java :

bean class.

```
public class Medium {
 int medId;
 String medName;

 /**
  * @return Returns the medId.
  */
 public int getMedId() {
  return medId;
 }
 /**
  * @param medId The medId to set.
  */
 public void setMedId(int medId) {
  this.medId = medId;
 }
 /**
  * @return Returns the medName.
  */
 public String getMedName() {
  return medName;
 }
 /**
  * @param medName The medName to set.
  */
```

```
 public void setMedName(String medName) {
  this.medName = medName;
 }
}
```

DAO Clas to retrive mediums in data base.

```
public static List  getMediums(){
  PreparedStatement pStmt = null;
     Connection conn = null;
     boolean success = false;
     ResultSet rs = null;
    List medList = new ArrayList();

     try{
      conn =  getConnection();

      String sql = " select * from MEDIUM ";
      pStmt = conn.prepareStatement(sql);

      rs = pStmt.executeQuery();
      while(rs.next()){
       Medium med = new Medium();
       med.setMedId(rs.getInt("MED_ID"));
       med.setMedName(rs.getString("MEDIUM_NAME"));
       medList.add(med);
      }



     }catch(Exception e){
      e.printStackTrace();

     }finally{
      closeConnectionProp(conn,pStmt,rs);
     }

     return medList;

 }
```

JSP :

```
<select name="Medium" id="medium" multiple=true>
      <option value="0">Choose A Medium</option>

       <%
   for(int i=0; i<medList.size();i++){
    Medium med = (Medium)medList.get(i);
  %>
      <option value="<%=med.getMedId()%>"><%=med.getMedName()%></option>
      <%}%>
     </select>
```

Servlet :
Retrive the multiple selections from jsp

```
String[] mediums = request.getParameterValues("medium");

for(int i=0;i<mediums.length;i++)
{
System.out.println(mediums[i]);
}
```

For Struts
How to create and retrive a drop down or selections list in Struts?
http://www.techfaq360.com/struts_interview_questions.jsp?qid=412

Question No : 8
Q.How to create a drop down list in jsp?
Answer :

This is the code to display drop down list in jsp.
List medList = DAO.getallMedium(); // this method retun list of Medium objects.
medList list contains list of Medium objects.

Java :
bean class.

```
public class Medium {
 int medId;
 String medName;

 /**
  * @return Returns the medId.
  */
 public int getMedId() {
  return medId;
 }
 /**
  * @param medId The medId to set.
  */
 public void setMedId(int medId) {
  this.medId = medId;
 }
 /**
  * @return Returns the medName.
  */
 public String getMedName() {
  return medName;
 }
 /**
  * @param medName The medName to set.
  */
 public void setMedName(String medName) {
  this.medName = medName;
 }
}
```

DAO Clas to retrive mediums in data base.

```
public static List  getMediums(){
  PreparedStatement pStmt = null;
     Connection conn = null;
     boolean success = false;
     ResultSet rs = null;
    List medList = new ArrayList();

     try{
      conn =  getConnection();

      String sql = " select * from MEDIUM ";
```

```
      pStmt = conn.prepareStatement(sql);

      rs = pStmt.executeQuery();
      while(rs.next()){
       Medium med = new Medium();
       med.setMedId(rs.getInt("MED_ID"));
       med.setMedName(rs.getString("MEDIUM_NAME"));
       medList.add(med);
      }



     }catch(Exception e){
      e.printStackTrace();

     }finally{
      closeConnectionProp(conn,pStmt,rs);
     }

     return medList;

 }
```

JSP :

```
<select name="Medium" id="medium" >
       <option value="0">Choose A Medium</option>

        <%
    for(int i=0; i<medList.size();i++){
     Medium med = (Medium)medList.get(i);
   %>
       <option value="<%=med.getMedId()%>"><%=med.getMedName()%></option>
       <%}%>
      </select>
```

Servlet :
Retrive the selection from jsp
String mediumId =(String)request.getParameter("medium");

Question No : 9

Q. How to add and delete Cookie in jsp ?
Answer :
Add Cookie to response object:

```
Cookie cookie = new Cookie ("name",value);
cookie.setPath("/");
cookie.setDomain(DOMAIN_NAME); DOMAIN_NAME may be .techfaq360.com
cookie.setMaxAge(2* 7 * 24 * 60 * 60);// 2 week
response.addCookie(cookie);
```

Get cookie from request object :

```
Cookie myCookie = null;

Cookie cookies [] = request.getCookies ();
if (cookies != null)
for (int i = 0; i < cookies.length; i++)
 {
 if (cookies [i].getName().equals ("name")) // the name of the cookie you have added
{
 myCookie = cookies[i];
 break;
 }
}
```

Delete Cookie:

You can't delete the cookie. just add maxage to 0;

```
cookie.setMaxAge(0);
cookie.setPath("/");
cookie.setDomain(DOMAIN_NAME);
response.addCookie(cookie);
```

cookie.setMaxAge(-1) means on browser close cookie will be deleted.

Question No : 10
Q. How to Protect JSPs from direct access ?
Answer :
JSPs located in the WEB-INF and its sub-directories are protected from outside access.
  If you want to go pageB.jsp from pageA.jsp

```
  <html:link action="gotoPageB">Go to Page B</html:link>
```

  in the struts-config.xml

```
  <action path="/gotoPageB"
parameter="/WEB-INF/pageB.jsp"
type="org.apache.struts.actions.ForwardAction"/>
```

Question No : 11

Q.How to upload an image from servlet/jsp into server from clients machine?

Answer :

Image upload from JSP to Data Base step by step :

step 1 :

In the JSP

```
<form name="regform2" method="post" enctype="multipart/form-data">
<input type="file" name="ImageFile" id="ImageFile" onChange="uploadImage()"/>
</form>
```

java script :

```
function uploadImage(){
document.regform.action ="<%=request.getContextPath()%>/dfdmin?cmd=uploadimage";
document.regform.submit();
}
```

Step 2.

In the servlet - add the below code.

This will upload your image to server (D:\\) or In unix you can mention /home/user like that

```
String rtempfile = File.createTempFile("temp","1").getParent();
MultipartRequest multi = new MultipartRequest(request, rtempfile, 500000 * 1024);
File rnewfile=null;
rnewfile = new
File("D:\\"+"jsp"+File.separator+"images"+File.separator+"uploadImage"+File.separator);
if(rnewfile.exists()){
}else{
rnewfile.mkdirs();
}

File f = multi.getFile("ImageFile");
System.out.println(f.getName());
FileInputStream fin =new FileInputStream(f);
RandomAccessFile r = new RandomAccessFile(rnewfile+File.separator+f.getName(),"rw");
filename = f.getName();
// FileOutputStream fos =new FileOutputStream(rnewfile);
```

```
byte sizefile[] = new byte[5000000];
fin.read(sizefile);
// fos.write(sizefile);
r.write(sizefile);
//fos.close();
r.close();
fin.close();

DAO.upload(f); // Call to DAO for insert image into database.
```

Step 3.

Then call to DAO to save this image to data base.
Insert into Database

```
public void upload(File f){
InputStream is = new FileInputStream(f);
String sql = " INSERT INTO image_upload (IMAGE) VALUES (?) ";
pStmt = conn.prepareStatement(sql);
pStmt.setBinaryStream(1, is, (int)(f.length()));
pStmt.execute();
conn.commit();
}
```

This will upload multipart file to your data base.

Note : get cos.jar from oreilly website

OR
if your server have already image file then just do below step

Code is here :

```
File f = new File("d:\\test.jpg");
InputStream is = new FileInputStream(f);
String sql = " INSERT INTO image_upload (IMAGE) VALUES (?) ";
pStmt = conn.prepareStatement(sql);
pStmt.setBinaryStream(1, is, (int)(f.length()));
```

```
pStmt.execute();
conn.commit();
```

This will work.

Question No : 12

Q.How do you delete a Cookie within a JSP?

Answer :

```
Cookie mycook = new Cookie("name","value");
response.addCookie(mycook);
Cookie killmycook = new Cookie("mycook","value");
killmycook.setMaxAge(0);
killmycook.setPath("/");
killmycook.addCookie(killmycook);
```

Question No : 13

Q.How do you prevent the Creation of a Session in a JSP Page and why?

Answer :

```
<%@ page session="false">
```

By default, a JSP page will automatically create a session for the request if one does not exist. However, sessions consume resources and if it is not necessary to maintain a session, one should not be created. For example, a marketing campaign may suggest the reader visit a web page for more information. If it is anticipated that a lot of traffic will hit that page, you may want to optimize the load on the machine by not creating useless sessions.

Question No : 14

Q.How does a servlet communicate with a JSP page?

Answer :

In the service method

```
protected void service(HttpServletRequest request,HttpServletResponse response)
  throws ServletException, java.io.IOException {

User user = (User)request.getSession().getAttribute("user");

  String imgId = (String)request.getParameter("imgId");
  String path = request.getContextPath()+"/jsp/addnetwork.jsp";

  //do some thing
            User user = DAO.getUser(imgId);
```

```
            request.getSession().setAttribute("user",user);
   response.sendRedirect(path);


}
```

In the addnetwork.jsp

```
User user = getAttribute("user);
```

Question No : 15

Q.which situation you use static include and dynamic include in jsp?

Answer :

static include : Contents are static like <%@ include file="header.jsp"%> ..Not changing frequently. Don't need to pass parameter. For example.

a.jsp contain <%@ include file="header.jsp"%> then all the codes of header.jsp is included in a.jsp.

dynamic include : Need to pass parameter. Need to execute every time. Dynamic content. Excecute the included jsp and only ouput is pasted to main jsp

For example..

a.jsp contain following code

```
<jsp:include page="masterTemplate.jsp" flush="true">
<jsp:param name="rColumn" value="rightDisplay.jsp" />
<jsp:param name="mColumn" value="AlertBody.jsp" />
<jsp:param name="title" value="Alerts" />
</jsp:include>
```

masterTemplate.jsp execute and output is pasted to a.jsp.

Question No : 16

Q.Difference between static include and dynamic include in JSP ?

Answer :

The syntax for static include is <%@ include file=?filename.jsp? %> and the syntax for dynamic include is <jsp:include page=?filename.jsp? />

Static include is an inline inclusion. i.e., the contents of the file will be included at translation phase. It?s something like a copy and paste .

In case of dynamic include the response of the file will be included to the original response at runtime. The file will be processed separately and only the response will become the part of the original files? response.

Static include cannot accept a parameter (What this parameter will do even if are passing it?). But dynamic include can accept a parameter. (Here we have some one to do something on the parameter).
Dynamic include

```
<jsp:include page="masterTemplate.jsp" flush="true">
<jsp:param name="rColumn" value="rightDisplay.jsp" />
<jsp:param name="mColumn" value="AlertBody.jsp" />
<jsp:param name="title" value="Alerts" />
</jsp:include>
```

we are passing parameters.

Question No : 17
q.What's the Difference between Forward and Include?
Answer :
The <jsp:forward> action enables you to forward the request to a static HTML file, a servlet, or another JSP.

```
<jsp:forward page="url" />
```

The JSP that contains the <jsp:forward> action stops processing, clears its buffer, and forwards the request to the target resource. Note that the calling JSP should not write anything to the response prior to the <jsp:forward> action.

You can also pass additional parameters to the target resource using the <jsp:param> tag.

```
<jsp:forward page="test.htm" >
  <jsp:param name="name1" value="value1" />
  <jsp:param name="name2" value="value2" />
</jsp:forward>
```

In this example, test.jsp can access the value of name1 using request.getParameter("name1").

<jsp:include execute the code and force a flush of the buffer in the output stream.
If a.jsp has the code like

```
<jsp:include page="template.jsp" flush="true" >
  <jsp:param name="name1" value="value1" />
</jsp:include>
```

then template.jsp execute and the output is placed in a.jsp

Question No : 18
Q.How do you pass control from one JSP page to another?

Answer :
Use the following ways to pass control of a request from one servlet to another or one jsp to another.
The RequestDispatcher object ?s forward method to pass the control.
Ex. <jsp:forward page="b.jsp" />
The response.sendRedirect method
Ex. response.sendRedirect("b.jsp");

Question No : 19
Q.How can I declare methods within my JSP page?

Answer :
In the declarion part
<%!
public int add(inti,intj){
return i+j;
}
%>

Question No : 20
Q.How can I implement a thread-safe JSP page?
Answer :
You can make your JSPs thread-safe by having them implement the SingleThreadModel interface. This is done by adding the directive <%@ page isThreadSafe="false" % > within your JSP page.

Question No : 21
Q.How does JSP handle run-time exceptions?
Answer :
You can use the errorPage attribute of the page directive to have uncaught runtime exceptions automatically forwarded to an error processing page.
For example:
redirects the browser to the JSP page error.jsp if an uncaught exception is encountered during request processing. Within error.jsp, if you indicate that it is an error-processing page, via the directive: isErrorPage=true.
the Throwable object describing the exception may be accessed within the error page via the

exception implicit object.

Question No : 22

Q.How do I perform browser redirection from a JSP page?

Answer :

You can use the response implicit object to redirect the browser to a different resource, as:

response.sendRedirect("http://www.exforsys.com/path/error.html");

You can also physically alter the Location HTTP header attribute, as shown below:

You can also use the:

Also note that you can only use this before any output has been sent to the client. I beleve this is the case with the response.sendRedirect() method as well. If you want to pass any paramateres then you can pass using >

Question No : 23

Q.How do I include static files within a JSP page?

Answer :

<%@ include file="header.jsp"%>

Static resources should always be included using the JSP include directive. This way, the inclusion is performed just once during the translation phase

---

Question No : 24

Q.How many JSP scripting elements and what are they?

Answer :

There are three scripting language elements:

--declarations

<%!

int i=8;

%>

--scriptlets

<% Java Code Logic

i=i+29;

%>

--expressions

<%= i %> - This will display the value of i.

Question No : 25

Q.What are the implicit objects?

Answer :

Implicit objects are objects that are created by the web container and contain information related

to a particular request, page, or application. They are:
--request
--response
--pageContext
--session
--application
--out
--config
--page
--exception

Question No : 26
Q.How does a try statement determine which catch clause should be used to handle an exception?

Answer :
When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exceptionis executed. The remaining catch clauses are ignored.

Question No : 27
Q.How do I prevent the browser from caching my dynamic content?

Answer :
add the code to your jsp.
<%
response.setHeader("Cache-Control", "no-cache");
response.setHeader("Pragma", "no-cache");
response.setIntHeader("Expires", 0);
%>

Question No : 28
Q.Explain the life-cycle mehtods in JSP?

Answer :
The jspInit()- The container calls the jspInit() to initialize te servlet instance.It is called before any other method, and is called only once for a servlet instance.
The _jspservice()- The container calls the _jspservice() for each request, passing it the request and the response objects.
The jspDestroy()- The container calls this when it decides take the instance out of service. It is the last method called n the servlet instance.

Question No : 29

Q.What are the different scope valiues for the <jsp:useBean>?

Answer :

Scopes are

page - with in the same page

request - after forward or include also you will get the request scope data.

session -

after senRedirect also you will get the session scope data. All data stored in session is available to end user till session closed or browser closed.

application -

Data will be available through out the application. One user can store data in application scope and other can get the data from application scope.

Question No : 30

What are implicit objects in JSP?

Answer :

Certain objects that are available for the use in jsp documents without being declared first. These objects are parsed by the JSP engine and inserted into the generated servlet. The implicit objects re listed below

request

response

pageContext

session

application

out

config

page

exception

Question No : 31

What is a Expression,Declaration,Scriptlet in jsp?

Answer :

expression

An expression tag contains a scripting language expression that is evaluated, converted to a String, and inserted where the expression appears in the JSP file. Because the value of an expression is converted to a String, you can use an expression within text in a JSP file. Like
<%= emp.getName()%>
<%= (new java.util.Date()).toLocaleString() %>
You cannot use a semicolon to end an expression.

All the expression code is converting to servlet. All the expressions go to inside service() method of the convert servlet.

Declaration
A declaration declares one or more variables or methods for use later in the JSP source file. A declaration must contain at least one complete declarative statement. You can declare any number of variables or methods within one declaration tag, as long as they are separated by semicolons. The declaration must be valid in the scripting language used in the JSP file.

<%! int i = 0; %>
<%! int a, b, c; %

You can add method to declaration part.

```
<%!
 public String trimData(String str){
   return str.trim();
}
%>
```

You can call the method within the jsp.

All the declaration code is converting to servlet. If you add method , the method is in convert servlet. All varibales are instance variable in the convert servlet.

Scriptlet
Scriptlet code is like java logic. you can declare varibales in the scriptlet and do the logic.
 All the Scriptlet go to inside service() method of the convert servlet.

```
<%
int n=10;
for(int i=0;i<n;i++){
}
 %>
```

Question No : 32
What is a Visible and Hidden Comment in jsp?

Answer :
On view source (browser) the Hidden Comment is not visible to end user.

On view source (browser) the Visible Comment is visible to end user.

Examples
<%-- This comment will not be visible in the page source --%>

<!-- This comment will visible in the page source -->

Question No : 33
Q. What is a Expression?

Answer :
An expression tag contains a scripting language expression that is evaluated, converted to a String, and inserted where the expression appears in the JSP file. Because the value of an expression is converted to a String, you can use an expression within text in a JSP file. Like
<%= someexpression %>
<%= (new java.util.Date()).toLocaleString() %>
You cannot use a semicolon to end an expression

****

****
Servlet Interview Questions
****
Question No : 1
Details about load on startup (load-on-startup tag) in Servlet ?
Answer :
Used to decide whether servlet will be " lazily " or " eagerly " loaded
Specified in web.xml file

If the value is not specified or is a Number < 0 then it means " lazy loading "

What " lazy loading " means is that servlet is NOT loaded by container on startup

Servlet in this case is loaded on the first client request - so the first client can experience poor performance

" Eager " loading means that the servlet is initialised on container startup

If there are two servelts A & B with values 0 & 1 than it means that Servlet A (having value = 0) will be loaded first

So if there are more than one servlet this element specifies the order of loading - lower integer values (including zero) are loaded first

If you specify this element but do not provide the value - even then the servlet will be the first servlet that gets loaded

To ensure that servlet follows " lazy loading " - do not provide this entry at all in web.xml file OR provide a negative number

---

Question No : 2

What's the difference between init() & init(ServletConfig) and Which is better ?

Answer :

Before start we have to understand the servlet flow.

For example you have servlet LoginServlet which extends HttpServlet

```
public class LoginServlet extends HttpServlet{
}
```

And your HttpServlet internally extends GenericServlet.

```
public abstract class GenericServlet
implements Servlet, ServletConfig, Serializable
{

public GenericServlet()
{
}



public void init()
throws ServletException
{
}
```

```
public ServletConfig getServletConfig()
{
return config;
}

public void init(ServletConfig config)
throws ServletException
{
this.config = config;
init();
}


public abstract void service(ServletRequest servletrequest, ServletResponse servletresponse)
throws ServletException, IOException;

private transient ServletConfig config;
}
```

Now servlet container flow

Step 1. Loads the servlet class and create instance of the servlet class (LoginServlet).
LoginServlet login = new LoginServlet();

Step 2. Then servlet container create ServletConfig object for that servlet and
Call login.init(ServletConfig);

Case 1 :
If you have overridden init(ServletConfig) method in your servlet then call goes to your init(ServletConfig) method .

```
public void init(ServletConfig config) throws ServletException {
        System.out.println("\n**** Initializing LoginServlet Init Servlet ********** \n");
super.init(config);
    }
```

It will print "Initializing LoginServlet Init Servlet" and call goes to supper class GenericServlet init(ServletConfig) method.

In the GenericServlet init(ServletConfig) method there is code
This.config= config // initialize the Servlet config object and it is available to you.

Case 2 :
If you overridden init() method and not overridden init(ServletConfig) method.
Servlet container call login.init(ServletConfig);

There is no method like init(ServletConfig) in your servlet so call directly goes to super class GenericServlet init(ServletConfig) method.
This.config= config
// initialize the Servlet config object and it is available to you.

You can get the Servlet config object using getServletConfig() method.

Conclusion: It is BETTER to use init(). If you use init() you have no such worries as calling super.init().

If you use init(servletconfig) and forgot to call super.init(config) then servletconfig object will not set and you will not get the servletconfig object.

Question No : 3
Servlet Life Cycle ?
Answer :
The life cycle of a servlet is controlled by the container in which the servlet has been deployed. When a request is mapped to a servlet, the container performs the following steps.

If an instance of the servlet does not exist, the Web container
1) Loads the servlet class.
2) Creates an instance of the servlet class.
Initializes the servlet instance by calling the init method. Initialization is covered in Initializing a Servlet.
3) Invokes the service method, passing a request and response object. Service methods are discussed in the section Writing Service Methods.
4)If the container needs to remove the servlet, it finalizes the servlet by calling the servlet's destroy method.

Details are here
[URL=http://techforum360.com/forum/viewthread?thread=35]http://techforum360.com/forum/viewt

hread?thread=35[/URL]

Question No : 4
Q. Can we override init() or init(ServletConfig) method of HttpServlet ? Which is better ?
Answer :
Before start we have to understand the servlet flow.
For example you have servlet LoginServlet which extends HttpServlet
public class LoginServlet extends HttpServlet{
}
And your HttpServlet internally extends GenericServlet.

```
public abstract class GenericServlet
implements Servlet, ServletConfig, Serializable
{

public GenericServlet()
{
}
public void init()
throws ServletException
{
}
public ServletConfig getServletConfig()
{
return config;
}
public void init(ServletConfig config)
throws ServletException
{
this.config = config;
init();
}

public abstract void service(ServletRequest servletrequest, ServletResponse servletresponse)
throws ServletException, IOException;

private transient ServletConfig config;
}
```

Now servlet container flow

Step 1. Loads the servlet class and create instance of the servlet class (LoginServlet).
LoginServlet login = new LoginServlet();

Step 2. Then servlet container create ServletConfig object for that servlet and
Call login.init(ServletConfig);

Case 1 :
If you have overridden init(ServletConfig) method in your servlet then call goes to your init(ServletConfig) method .

```
public void init(ServletConfig config) throws ServletException {
        System.out.println("\n**** Initializing LoginServlet Init Servlet ********** \n");
 super.init(config);
    }
```

It will print "Initializing LoginServlet Init Servlet" and call goes to supper class GenericServlet init(ServletConfig) method.

In the GenericServlet init(ServletConfig) method there is code
This.config= config // initialize the Servlet config object and it is available to you.

Case 2 :
If you overridden init() method and not overridden init(ServletConfig) method.

Servlet container call login.init(ServletConfig);

There is no method like init(ServletConfig) in your servlet so call directly goes to super class GenericServlet init(ServletConfig) method.

This.config= config // initialize the Servlet config object and it is available to you.

You can get the Servlet config object using getServletConfig() method.

Conclusion: It is BETTER to use init(). If you use init() you have no such worries as calling super.init().
If you use init(servletconfig) and forgot to call super.init(config) then servletconfig object will not set and you will not get the servletconfig object.

More about this on
[URL=http://techforum360.com/forum/viewthread?thread=31]http://techforum360.com/forum/viewthread?thread=31[/URL]

Question No : 5
Q. Can we override service method in HttpServlet ?

Answer :
You can override service method in HttpServlet also.
If you override service method in HttpServlet then call will go to service() method instead of doGet() or doPost() method.

If the jsp form you mentioned
<form name="reg" method="post">
then also call will go to service method of the servlet. Call don't go to doPost() method. you can call doPost() method explicitly from servive() method.
If the jsp form you mentioned
<form name="reg" method="get">
then also call will go to service method of the servlet. Call don't go to doGet() method. you can call doGet () method explicitly from servive() method.

Question No : 6
Q.What is the difference between init() and init(ServletConfig) ?
Answer :
Before start we have to understand the servlet flow.
For example you have servlet LoginServlet which extends HttpServlet
public class LoginServlet extends HttpServlet{
}
And your HttpServlet internally extends GenericServlet.

```
public abstract class GenericServlet
    implements Servlet, ServletConfig, Serializable
{

    public GenericServlet()
    {
    }



    public void init()
        throws ServletException
    {
    }
```

```
    public ServletConfig getServletConfig()
    {
        return config;
    }

    public void init(ServletConfig config)
        throws ServletException
    {
        this.config = config;
        init();
    }


    public abstract void service(ServletRequest servletrequest, ServletResponse servletresponse)
        throws ServletException, IOException;

    private transient ServletConfig config;
}
```

Now servlet container flow

Step 1. Loads the servlet class and create instance of the servlet class (LoginServlet).
LoginServlet login = new LoginServlet();

Step 2. Then servlet container create ServletConfig object for that servlet and
Call login.init(ServletConfig);

Case 1 :
If you have overridden init(ServletConfig) method in your servlet then call goes to your init(ServletConfig) method .

```
public void init(ServletConfig config) throws ServletException {
  System.out.println("\n**** Initializing LoginServlet Init Servlet ********** \n"); super.init(config);
 }
```

It will print "Initializing LoginServlet Init Servlet" and call goes to supper class GenericServlet init(ServletConfig) method.

In the GenericServlet init(ServletConfig) method there is code
This.config= config // initialize the Servlet config object and it is available to you.

Case 2 :
If you overridden init() method and not overridden init(ServletConfig) method.
Servlet container call login.init(ServletConfig);

There is no method like init(ServletConfig) in your servlet so call directly goes to super class GenericServlet init(ServletConfig) method.
 This.config= config
// initialize the Servlet config object and it is available to you.

You can get the Servlet config object using getServletConfig() method.

Conclusion: It is BETTER to use init(). If you use init() you have no such worries as calling super.init().

If you use init(servletconfig) and forgot to call super.init(config) then servletconfig object will not set and you will not get the servletconfig object.

Question No : 7
Q.How to upload an File/image from servlet/jsp into server from clients machine?
Answer :
In the JSP

<form name="regform2" method="post" enctype="multipart/form-data">

<input type="file" name="ImageFile" id="ImageFile" onChange="uploadImage()"/>

</form>

uploadImage() - java script - document.regform.action ="<%=request.getContextPath()%>/dfdmin?cmd=uploadimage";

document.regform.submit();

In the servlet - add the below code.

String rtempfile = File.createTempFile("temp","1").getParent();

MultipartRequest multi = new MultipartRequest(request, rtempfile, 500000 * 1024);

```
        File rnewfile=null;
        rnewfile = new
File("D:\\"+"jsp"+File.separator+"images"+File.separator+"uploadImage"+File.separator);

        if(rnewfile.exists()){

        }else{

           rnewfile.mkdirs();

        }
        //Enumeration files = multi.getFileNames();
        //while (files.hasMoreElements()) {
        File f = multi.getFile("ImageFile");
        System.out.println(f.getName());
        FileInputStream fin =new FileInputStream(f);        RandomAccessFile r = new
RandomAccessFile(rnewfile+File.separator+f.getName(),"rw");
         filename = f.getName();
         // FileOutputStream fos =new FileOutputStream(rnewfile);
         byte sizefile[] = new byte[5000000];
         fin.read(sizefile);
         // fos.write(sizefile);
         r.write(sizefile);
         //fos.close();
         r.close();
        fin.close();
```

This will upload multipart file to your path.

Note : get cos.jar from oreilly website.

Question No : 8

What's the difference between init() & init(ServletConfig) in genericServlet ?

Answer :

init(ServletConfig):

The default implementation of init(ServletConfig) does some initialization then calls init(). If you use init(ServletConfig) and forget to call super.init(config) at the start of the method then your

Servlet will not be initialized correctly.
Called by the servlet container to indicate to a servlet that the servlet is being placed into service.
You will get the ServletConfig object.

init():
You can get the ServletConfig using getServletConfig().
A convenience method which can be overridden so that there's no need to call super.init(config).
Instead of overriding init(ServletConfig), simply override this method and it will be called by GenericServlet.init(ServletConfig config). The ServletConfig object can still be retrieved via getServletConfig().

It is BETTER to use init(). If you use init() you have no such worries as calling super.init().

Details :

Before start we have to understand the servlet flow.
For example you have servlet LoginServlet which extends HttpServlet
public class LoginServlet extends HttpServlet{
}
And your HttpServlet internally extends GenericServlet.

```
public abstract class GenericServlet
    implements Servlet, ServletConfig, Serializable
{

    public GenericServlet()
    {
    }


    public void init()
        throws ServletException
    {
    }


    public ServletConfig getServletConfig()
    {
        return config;
    }
```

```
    public void init(ServletConfig config)
        throws ServletException
    {
        this.config = config;
        init();
    }


    public abstract void service(ServletRequest servletrequest, ServletResponse servletresponse)
        throws ServletException, IOException;

    private transient ServletConfig config;
}
```

Now servlet container flow

Step 1. Loads the servlet class and create instance of the servlet class (LoginServlet).
LoginServlet login = new LoginServlet();

Step 2. Then servlet container create ServletConfig object for that servlet and
Call login.init(ServletConfig);

Case 1 :
If you have overridden init(ServletConfig) method in your servlet then call goes to your init(ServletConfig) method .

```
public void init(ServletConfig config) throws ServletException {
  System.out.println("\n**** Initializing LoginServlet Init Servlet ********** \n"); super.init(config);
 }
```

It will print "Initializing LoginServlet Init Servlet" and call goes to supper class GenericServlet init(ServletConfig) method.

In the GenericServlet init(ServletConfig) method there is code
This.config= config // initialize the Servlet config object and it is available to you.

Case 2 :
If you overridden init() method and not overridden init(ServletConfig) method.
Servlet container call login.init(ServletConfig);

There is no method like init(ServletConfig) in your servlet so call directly goes to super class GenericServlet init(ServletConfig) method.
This.config= config
// initialize the Servlet config object and it is available to you.

You can get the Servlet config object using getServletConfig() method.

Conclusion: It is BETTER to use init(). If you use init() you have no such worries as calling super.init().

If you use init(servletconfig) and forgot to call super.init(config) then servletconfig object will not set and you will not get the servletconfig object.

Question No : 9
Q.What is the difference between ServletContext and PageContext?

Answer :
ServletContext: Gives the information about the container
PageContext: Gives the information about the Request

Question No : 10
Q.What is the difference in using request.getRequestDispatcher() and context.getRequestDispatcher()?

Answer :
The difference between ServletRequest.getRequestDispatcher(String path) and ServletContext.getRequestDispatcher(String path) is that the former can accept a relative path as well whereas the latter can accept paths relative to the current context root only.

If the path starts with a '/' in the getRequestDispatcher(String path) of the ServletRequest interface then it's interpreted as being relative to the current context root otherwise it'll be a relative to the request of the calling servlet. Whereas, the path of the getRequestDispatcher(String path) of the ServletContext interface must start with '/' only - being relative to the current context root.

Another difference between the two is that path of the getRequestDispatche(String path) of the

ServletRequest interface cannot extend outside the current servlet context whereas getRequestDispatcher(String path) of the ServletContext can use the getContext(String uripath) method to obtain RequestDispatcher for resources in foreign contexts.

get more details on
[URL=http://techforum360.com/forum/viewthread?thread=28]http://techforum360.com/forum/viewthread?thread=28[/URL]

Question No : 11

Q.Difference forward() and response.sendRedirect() .

Answer :

difference between the two is that sendRedirect always sends a header back to the client/browser. this header then contains the resource(page/servlet) which u wanted to be redirected. the browser uses this header to make another fresh request. thus sendRedirect has a overhead as to the extra remort trip being incurred. its like any other Http request being generated by ur browser. the advantage is that u can point to any resource(whether on the same domain or some other domain). for eg if sendRedirect was called at www.mydomain.com then it can also be used to redirect a call to a resource on www.techfaq360.com.

where as in case of forward() call, the above is not true. resources from the server, where the fwd. call was made, can only be requested for. but the major diff between the two is that forward just routes the request to the new resources which u specify in ur forward call. that means this route is made by the servlet engine at the server level only. no headers r sent to the browser which makes this very eficient. also the request and response objects remain the same both from where the forward call was made and the resource which was called.

In case of forward you can get the data from request object but in case of sendRedirect() you can get from session object, request object is not avilable.

Question No : 12

Q.Can we use the constructor, instead of init(), to initialize servlet?

Answer :

Yes , of course you can use the constructor instead of init(). There's nothing to stop you. The original reason for init() was that ancient versions of Java couldn't dynamically invoke constructors with arguments, so there was no way to give the constructur a ServletConfig. That no longer applies, but servlet containers still will only call your no-arg constructor. So you won't have access to a ServletConfig or ServletContext.

Question No : 13
Q.Explain the life cycle of Servlet?
Answer :
The life cycle of a servlet is controlled by the container in which the servlet has been deployed. When a request is mapped to a servlet, the container performs the following steps.

If an instance of the servlet does not exist, the Web container
1) Loads the servlet class.
2) Creates an instance of the servlet class.
Initializes the servlet instance by calling the init method. Initialization is covered in Initializing a Servlet.
3) Invokes the service method, passing a request and response object. Service methods are discussed in the section Writing Service Methods.
4)If the container needs to remove the servlet, it finalizes the servlet by calling the servlet's destroy method.

Details are here
[URL=http://techforum360.com/forum/viewthread?thread=35]http://techforum360.com/forum/viewthread?thread=35[/URL]

Question No : 14
Q.Why don't we write a constructor in a servlet?
Answer :
Container writes a no argument constructor for our servlet.

Question No : 15
Q.When we don't write any constructor for the servlet, how does container create an instance of servlet?
Answer :
Container creates instance of servlet by calling Class.forName(className).newInstance().

Question No : 16
Q.Why is it that we can't give relative URL's when using ServletContext.getRequestDispatcher() when we can use the same while calling ServletRequest.getRequestDispatcher()?
Answer :
Since ServletRequest has the current request path to evaluae the relative path while ServletContext does not.

Question No : 17
Q.Once the destroy() method is called by the container, will the servlet be immediately destroyed?

What happens to the tasks(threads) that the servlet might be executing at that time?
Answer :
Yes, but Before calling the destroy() method, the servlet container waits for the remaining threads that are executing the servlet?s service() method to finish.

Question No : 18
Q.Request parameter How to find whether a parameter exists in the request object?

Answer :
1.boolean hasFoo = !(request.getParameter("foo") == null ||
request.getParameter("foo").equals(""));
2. boolean hasParameter = request.getParameterMap().contains(theParameter);

Question No : 19
What is new in ServletRequest interface ?
Answer :
The following methods have been added to ServletRequest 2.4 version:
public int getRemotePort()
public java.lang.String getLocalName()
public java.lang.String getLocalAddr()
public int getLocalPort()

Question No : 20
Q.Given the request path below, which are context path, servlet path and path info?
/bookstore/education/index.html
Answer :
context path: /bookstore
servlet path: /education
path info: /index.html

Question No : 21
Q.When a client request is sent to the servlet container, how does the container choose which servlet to invoke?
Answer :
The servlet container determines which servlet to invoke based on the configuration of its servlets, and calls it with objects representing the request and response.
For Example---
<servlet>
<servlet-name>admin</servlet-name>
<servlet-class>com.servlet.LoginServlet</servlet-class>
</servlet>

```
<servlet-mapping>
 <servlet-name>
 admin
 </servlet-name>
 <url-pattern>
 /artadmin
 </url-pattern>
  </servlet-mapping>
```

if the url in browser is
http://localhost:8080/artadmin

then server will call LoginServlet. Based on the above mapping.

Question No : 22
Q.What is servlet container?

Answer :
The servlet container is a part of a Web server or application server that provides the network services over which requests and responses are sent, decodes MIME-based requests, and formats MIME-based responses. A servlet container also contains and manages servlets through their lifecycle.

Question No : 23
Why IllegalStateException in jsp/servet?
Answer :
by attempting to redirect from within the middle of a JSP page you will get IllegalStateException .
For Example

```
<div>News </div>
<%
 if ("not logged in")
        response.sendRedirect("login.jsp");
%>
<div>more news</div>
```

You can avoid this bu using return ; Example :

```
<div>News </div>
<%
 if ("not logged in")
        response.sendRedirect("login.jsp");
        return;
```

```
%>
<div>more news</div>

or
public void doGet(HttpServletRequest  request,
                HttpServletResponse response) throws ServletException, IOException{

    if("client".equals(request.getParameter("user_type"))){
        response.sendRedirect("client-screen.jsp");
        return;  // <-------  This return statement prevents any further writing to the outputStream
    }

    //
    // Other code that may write to the outputStream....
    //

}
```

Question No : 24

Q.How do I upload a file in a servlet app?

Answer :

In the JSP

```
<form name="regform2" method="post"  enctype="multipart/form-data">
        <input type="file" name="ImageFile" id="ImageFile"  onChange="uploadImage()"/>
  </form>
```

uploadImage() - java script - document.regform.action

```
="<%=request.getContextPath()%>/dfdmin?cmd=uploadimage";
document.regform.submit();

In the servlet - add the below code.

String rtempfile = File.createTempFile("temp","1").getParent();
  MultipartRequest multi = new MultipartRequest(request, rtempfile, 500000 * 1024);
  File rnewfile=null;
  rnewfile = new
File(CommonArt.IMAGE_PATH+"jsp"+File.separator+"images"+File.separator+"uploadImage"+Fil
```

```
e.separator);


  if(rnewfile.exists()){

  }else{

   rnewfile.mkdirs();
  }
  //Enumeration files = multi.getFileNames();
  //while (files.hasMoreElements())  {
  File f = multi.getFile("ImageFile");
  System.out.println(f.getName());
  FileInputStream fin =new FileInputStream(f);
  RandomAccessFile r = new RandomAccessFile(rnewfile+File.separator+f.getName(),"rw");
     filename = f.getName();
    // FileOutputStream fos =new  FileOutputStream(rnewfile);
     byte sizefile[] = new byte[5000000];
     fin.read(sizefile);
    // fos.write(sizefile);
     r.write(sizefile);
     //fos.close();
    r.close();
  fin.close();
```

This will upload multipart file to your path.

Note : get cos.jar from oreilly website.

Question No : 25

Q.What's the difference between response.sendRedirect() and requestDispatcher.forward() ?

Answer :

response.sendRedirect() :

This is complete new request to browser.

Request is not maintained so data stored in request object not avilable to redirect page.

You have to store data in session to get in the reditect jsp.

You can pass data like response.sendRedirect("/techfaq360.jsp?name=satya). then in the techfaq360.jsp you have to get name using request.getParameter("name");

This is fresh call to server.

For example .. you forwarded from /login to login.jsp
in the browser you can see /login.jsp....

requestDispatcher.forward() : This not new request. Just transfer the content to forwarded jsp.
You can store data in request and able to get in redirect jsp.

For example .. you forwarded from /login to login.jsp
in the browser you can see /login ....

Disadvantage - browser refresh call to /loginj again. may be duplicate data submit.

Question No : 26
Q.How do I implement security for my web application ?
Answer :
The use of HTTPS/SSL can be required by adding the following to the web.xml file:

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>Entire Application</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>

    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

Question No : 27
Q.How can I use servlets with protocols other than HTTP, e.g. FTP?
Answer :
The javadocs for javax.servlet.Servlet and GenericServlet make it sound as if protocols other than HTTP can be used simply by extending GenericServlet, and implementing the methods that deal

with the protocol, much like HttpServlet does for HTTP. That is not the case. The protocol needs to be supported by the servlet engine (which does all the network handling for the servlet), and no servlet engine exists that supports anything other than HTTP(S). Adding a different protocol would be a big project, especially since other protocols don't have the same request/response nature of HTTP. If you find yourself contemplating such a project, post your requirements to the Servlet forum, and a better solution will probably be suggested.

For JSPs, the specification says "Most JSP pages use the HTTP protocol, but other protocols are allowed by this specification.". Since a non-HTTP JSP implementation would depend on a non-HTTP servlet implementation, that too is just a theoretical possibility.

(Note that all of this has nothing to do with a servlet's ability to be a client for other protocols. E.g., by using the JavaMail or Jakarta Commons Net APIs a servlet can access SMTP and FTP servers without problems.)

Question No : 28
Q.What happens if i call destroy() from init()?

Answer :
Destroy the servlet as soon as start.

Question No : 29
Q.Why can't a container call constructor having parameters?
Answer :
As it is the container that manages a servlets lifecycle, you must define a generic way of working for all servlets. You can't use the constructor because otherwise you would have to modify the container to tell him to instantiate this particular servlet

---

Question No : 30
Q.Why do servlets have an init method? Can't we make use of the servlet constructor for initialization?
Answer :
You can't make use of the constructor because the container calls it and therefore you can't pass any parameters to the constructor. Also at the point the constructor is called the class is not really a Servlet because it doesn't have a reference to the ServletConfig, which provides all the initialisation parameters etc.

Because the servlet container manages the servlet lifecycle, one should never call the constructor, the init and destroy methods.

Question No : 31
Q.Explain the life cycle methods of a Servlet.

Answer :
The javax.servlet.Servlet interface defines the three methods known as life-cycle method.
public void init(ServletConfig config) throws ServletException
public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException
public void destroy()
First the servlet is constructed, then initialized wih the init() method.
Any request from client are handled initially by the service() method before delegating to the doXxx() methods in the case of HttpServlet.

The servlet is removed from service, destroyed with the destroy() methid, then garbaged collected and finalized.

Question No : 32
Q.Explain the directory structure of a web application
Answer :
The directory structure of a web application consists of two parts.
A private directory called WEB-INF
A public resource directory which contains public resource folder.

WEB-INF folder consists of
1. web.xml
2. classes directory
3. lib directory

Question No : 33
Q.What are the common mechanisms used for session tracking?

Answer :
1) Cookies - Must be cookie on to your browser
http session object - it used jsession id internally
URL- rewriting - append variables to url
Hidden variable - you can use hidden variable to track

all methods just send some value to server every time and check it is same or not. if it is same then session maintains.

Question No : 34
What is load-on-startup in servlet ?
Answer :
On server startup servlet will be called by the server.
the value 1 in <load-on-startup>1</load-on-startup> specifies the order in which the servlet must be loaded by the server..

if you have 2 servlet you can specify <load-on-startup>2</load-on-startup> for the next servlet

Below is the servlet tag entry in web.xml:

<servlet>
<servlet-name>InitServlet</servlet-name>
<servlet-class>util.InitServlet</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet

Question No : 35
Q. What is the difference between the getRequestDispatcher(String) and getNamedDispatcher(String) methods in the ServletContext Class?
Answer :
NamedDispatcher

Returns a RequestDispatcher object that acts as a wrapper for the named servlet.

getNamedDispatcher(String) method takes the name of the Servlet as parameter which is declared via Deployment descriptor.

Example: Deployment Descriptor

<servlet>
    <servlet-name>ServletTest</servlet-name>
    <servlet-class>com.example.ServletTest</servlet-class>
</servlet>

RequestDispatcher dispatch = request.getNamedDispatcher("ServletTest");
dispatch.forward(request, response);
Note: A servlet instance can determine its name using servletConfig.getServletName(); This

method returns the name of the class that implements the Servlet interface or extends the HttpServlet class.

RequestDispatcher

Returns a RequestDispatcher object that acts as a wrapper for the resource located at the given path.

```
 RequestDispatcher dispatch = request.getRequestDispatcher("/satya");
```
Here "/satya" represents the url-pattern element value of the servlet class.

```
<servlet-mapping>
    <servlet-name>Test</servlet-name>
    <url-pattern>/satya</url-pattern>
</servlet-mapping>
```

get details about difference between the getRequestDispatcher(String path) method of ServletRequest interface and ServletContext interface?

[URL=http://techforum360.com/forum/viewthread?thread=28]http://techforum360.com/forum/viewthread?thread=28[/URL]

Question No : 36
Q: What is the difference between Difference between doGet() and doPost()?

Answer :
difference between Difference between doGet() and doPost() --
1) by default httpservlet will call doGet method
   if you mention method=post then only doPost() method will be called.

2) doGet() method is limited with 2k of data to be sent, and doPost() method doesn't have this limitation

3) if your action is
http://techfaq360.com/adminservlet
and method = get then action will be like

http://techfaq360.com/adminservlet?cmd=test&name=satya ..all the form data you can see on address bar.

In case of doPost you can't see the data.

Question No : 37

Q. What is the difference between ServletContext and ServletConfig?

Answer :

I have mentioned some tips regarding ServletContext and ServletConfig. okey Do well

ServletContext Defines a set of methods that a servlet uses to communicate with its servlet container.

ServletConfig is a servlet configuration object used by a servlet container used to pass information to a servlet during initialization. All of its initialization parameters can ONLY be set in deployment descriptor.

The ServletContext object is contained within the ServletConfig object, which the Web server provides the servlet when the servlet is initialized.

You can specify param-value pairs for ServletContext object in <context-param> tags in web.xml file.

The ServletConfig parameters are specified for a particular servlet and are unknown to other servlets.

The ServletContext parameters are specified for an entire application outside of any particular servlet and are available to all the servlets within that application.

Question No : 38

Q: What is the difference between HttpServlet and GenericServlet?

Answer :

Difference are

1) HttpServlet extends GenericServlet

HttpServlet Provides an abstract class to be subclassed to create an HTTP servlet suitable for a Web site. A subclass of HttpServlet must override at least one method, usually one of these:

doGet, if the servlet supports HTTP GET requests

doPost, for HTTP POST requests
doPut, for HTTP PUT requests
doDelete, for HTTP DELETE requests
init and destroy, to manage resources that are held for the life of the servlet
getServletInfo, which the servlet uses to provide information about itself
There?s almost no reason to override the service method. service handles standard HTTP requests by dispatching them to the handler methods for each HTTP request type (the doXXX methods listed above). Likewise, there?s almost no reason to override the doOptions and doTrace methods.

GenericServlet defines a generic, protocol-independent servlet. To write an HTTP servlet for use on the Web, extend HttpServlet instead.

GenericServlet implements the Servlet and ServletConfig interfaces. GenericServlet may be directly extended by a servlet, although it?s more common to extend a protocol-specific subclass such as HttpServlet.

GenericServlet makes writing servlets easier. It provides simple versions of the lifecycle methods init and destroy and of the methods in the ServletConfig interface. GenericServlet also implements the log method, declared in the ServletContext interface.

To write a generic servlet, you need only override the abstract service method.

Question No : 39
Q.What is the difference between the getRequestDispatcher(String path) method of ServletRequest interface and ServletContext interface?

Answer :
The servletRequest's getRequestDispatcher() can take a relative path while ServletContext's getRequestDispatcher() can not(can only take relative to the current context's root). For example with ServletContext both request.getRequestDispatcher("./jsp/jsppage.jsp") - evaluated relative to the path of the request request.getRequestDispatcher("/jsp/jsppage.jsp") - evaluated relative to the root are all valid with ServletContext only context.getRequestDispatcher("/jsp/jsppage.jsp") is valid but not context.getRequestDispatcher("./jsp/jsppage.jsp"). that is it can not evaluate a path other than context root.

get details

[URL=http://techforum360.com/forum/viewthread?thread=28]http://techforum360.com/forum/viewthread?thread=28[/URL]

****

****

Struts Interview Questions

****

Question No : 1

Q.What is Action Class? Explain with Example ?

Answer :

An Action class in the struts application extends Struts 'org.apache.struts.action.Action" Class.
Action class acts as wrapper around the business logic and provides an inteface to the application's Model layer.
An Action works as an adapter between the contents of an incoming HTTP request and the business logic that corresponds to it.
Then the struts controller (ActionServlet) slects an appropriate Action and Request Processor creates an instance if necessary,
and finally calls execute method of Action class.
To use the Action, we need to Subclass and overwrite the execute() method. and your bussiness login in execute() method.
The return type of the execute method is ActionForward which is used by the Struts Framework to forward the request to the JSP as per the value of the returned ActionForward object.
ActionForward JSP from struts_config.xml file.

Developing our Action Class :

Our Action class (EmpAction.java) is simple class that only forwards the success.jsp.
Our Action class returns the ActionForward called "success", which is defined in the struts-config.xml file (action mapping is show later in this page).
Here is code of our Action Class

```
public class EmpAction extends Action
{
  public ActionForward execute(
    ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
      return mapping.findForward("success");
  }
}
```

mapping.findForward("success"); forward to JSP mentioned in struts_config.xml.
struts_config.xml configuration is :

```
  <action
     path="/EmpAction"
     type="com.techfaq.EmpAction">
     <forward name="success" path="/success.jsp"/>
   </action>
```

mapping.findForward("success") method forward to success.jsp (mentioned in struts_config.xml);

Here is the signature of the execute() method Action Class.

```
public ActionForward execute(ActionMapping mapping,
                             ActionForm form,
                             javax.servlet.http.HttpServletRequest request,
                             javax.servlet.http.HttpServletResponse response)
                      throws java.lang.Exception
```

Where
mapping - The ActionMapping used to select this instance
form - The optional ActionForm bean for this request (if any)
request - The HTTP request we are processing
response - The HTTP response we are creating
Throws:
Action class throws java.lang.Exception - if the application business logic throws an exception

In the browser : http://localhost:8080/testApp/EmpAction.do
This will call to execute() method of EmpAction and after that based on
mapping.findForward("success") forward to success.jsp.

Setup your first Action class
[URL=http://www.techfaq360.com/tutorial/struts_setup.jsp]http://www.techfaq360.com/tutorial/struts_setup.jsp[/URL]

Question No : 2
Q.How you can do Exception Handling in Struts ?
Answer :
There are two approaches available for the exception handling in struts.

Declarative:
Exceptions are defined in the struts-config.xml file and
if the exception occurs the control is automatically passed to the appropriate error page.
The <exception> tag is used to define the exception in the struts-config.xml file.

For Example : ((If RuntimeException in SaveEmpAaction class , control goes to exception.jsp)

```
<action path="/saveEmp"
type="com.techfaq.SaveEmpAaction"
input="/empform.jsp" >
    <exception key="error.system" type="java.lang.RuntimeException" path="/exception.jsp" />
</action>
```

Where

Key: The key defines the key present in MessageResources.properties file to describe the exception occurred.

Type: The class of the exception occurred.

Path: The page where the control is to be followed in case exception occurred.

Handler: The exception handler which will be called before passing the control to the file specified in path attribute

OR

Defining the Exceptions Globally for the struts-config.xml : (If RuntimeException in any Action class , control goes to exception.jsp)

```
<global-exceptions>
    <exception key="error.system" type="java.lang.RuntimeException" path="/exception.jsp" />
</global-exceptions>
```

Programmatic:
In this approach the exceptions are caught using normal java language try/catch block and instead of showing the exception some meaningful messages are displayed.
In this approach the flow of control is also maintained by the programs.
The main drawback of the approach is the developer has to write the code for the flow of the

application.

Question No : 3

Q. What are the Advantages of Struts ?

Answer :

Struts follow MVC framework. So the JSP, Java and Action classes are organized and easily maintainable.

Struts offers many advantages to the application programmer while reducing the development time and making the manageability of the application easier.

Advantages of Struts :

Centralized Configuration :

Rather than hard coding information into java programs,many Struts values are represented in XML or property files.

Struts_config.xml file is the place from where you can get all information?s about your web application. This is organized.

Your Action class , Form bean and JSP page information is in Struts_config.xml so don't need to search . All info in one place.

Form Beans :

Don't need to set the form vales to your value object . When you want to capture data from a form (In the servlet you do request.getParameter()).

In the struts you don't need to do explicitly request.getParameter(). Struts request processor will do for you. All the input data will be set to form bean.

Bean Tags :

Struts provides a set of custom JSP tags (bean:write,in particular) that let you easily output the properties of JavaBeans components.

Basically,these are concise and powerful variations of the standard jsp:useBean and jsp:getProperty tags.

HTML tags :

Struts provides a set of custom JSP tags to create HTML forms that are associated with JavaBeans components.

Form Field Validation :

Apache Struts has built-in capabilities for checking that form values are in the required format.
If values are missing or in an improper format,the form can be automatically redisplayed with error messages and with the previously entered values maintained.
This validation can be performed on the server (in Java),or both on the server and on the client (in JavaScript).

Question No : 4
How Iterate Tag used with a Map ?

Answer :

```
<logic:iterate id="user" name="userForm" property="usermap">
  <tr>
    <td>User Id: <bean:write name=" user" property="key"/></td>
    <td>Password: <bean:write name=" user" property="value"/></td>
  </tr>
</logic:iterate>
```

Question No : 5
Q.How does client side validation using validator framework work in struts ?
Answer :
There are two configuration files used.
One if called validator-rules.xml and the other is validation.xml.
This is example of client side validation :Java Script message.

Step 1.
In the JSP page : empform.jsp

```
<html:form action="/EmpSaveAaction" method="post" onsubmit="return validateEmpForm(this);">

<html:text property="firstName" size="30" maxlength="30"/>
<html:text property="lastName" size="30" maxlength="30"/>

<html:submit>Save</html:submit>

<!-- Begin Validator Javascript Function-->
<html:javascript formName="empForm"/>
<!-- End of Validator Javascript Function-->
</html:form>
```

You have to add
<html:javascript formName="empForm"/> in JSP for activate client side validation .

Step 2.
Add action mapping in struts-config.xml

```
<action
    path="/EmpSaveAaction"
    type="empForm"
    name="AddressForm"
    scope="request"
    validate="true"
    input="/empform.jsp">
    <forward name="success" path="/success.jsp"/>
</action>
```

and add the form bean inside <form-beans> </form-beans> tag

```
<form-beans>
<form-bean name="empForm" type="com.techfaq.form.EmpForm" />
 </form-beans>
```

Step 3.
validator-rules.xml :

The validator-rules.xml file defines the Validator definitions available for a given application. The validator-rules.xml file acts as a template, defining all of the possible Validators that are available to an application.
Example validator-rules.xml File :

```
<form-validation>
 <global>
  <validator
    name="required"
    classname="org.apache.struts.util.StrutsValidator"
    method="validateRequired"
    methodparams="java.lang.Object,
                  org.apache.commons.validator.ValidatorAction,
                  org.apache.commons.validator.Field,
                  org.apache.struts.action.ActionErrors,
                  javax.servlet.http.HttpServletRequest"
    msg="errors.required"/>
```

```
  </global>
</form-validation>
```

Step 4.
validation.xml File :
The validation.xml file is where you couple the individual Validators defined in the validator-rules.xml to components within your application.
validation.xml File

```
<form-validation>
 <formset>
  <form name="empForm">
    <field
      property="firstName"
      depends="required">
      <arg key="label.firstName"/>
    </field>

    <field
      property="lastName"
      depends="required">
      <arg key="label.lastName"/>
    </field>
  </form>
 </formset>
</form-validation>
```

In the empForm firstName and lastName are the required filed.
So in the above configuration you can see we add for both firstName and lastName.
You can see depends="required" - "required" property is defind in validator-rules.xml.

In the resource bundle : application_resource.propertis file
label.firstName=First Name
label.lastName=Last Name

```
#Error messages used by the Validator
errors.required={0} is required.
```

Based on the validation.xml File configuration .
Error in jsp will be : Java Script message.
First Name is required.
Last Name is required.

{0} will be filled by (First Name or Last Name) because validation.xml above configuration you have defind <arg key="label.firstName"/>.

Question No : 6
Q. How to do File Upload in Struts ?
Answer :
Step 1.
Create a form bean

```
public class FileUploadForm extends ActionForm
{
  private FormFile file;

  public FormFile getFile() {
    return file;
  }

  public void setFile(FormFile file) {
    this.file = file;
  }
}
```

Step 2.

In the struts-config.xml file add

```
<form-bean
name="FileUploadForm"
type="com.techfaq.form.FileUploadForm"/>
```

Step 3.

add action mapping entry in the struts-config.xml file:

```
<action
path="/FileUploadAndSave"
type="com.techfaq.action.FileUploadAndSaveAction"
name="FileUploadForm"
scope="request"
```

validate="true"
input="/pages/fileupload.jsp">
<forward name="success" path="/jsp/success.jsp"/>
</action>

Step 4.
In the JSP
<html:form action="/FileUploadAndSave" method="post" enctype="multipart/form-data">
File Name

<html:file property="file"/>

<html:submit>Upload File</html:submit>

</html:form>

Step 5.
In the Action class write the code

```
public ActionForward execute(
    ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    FileUploadForm myForm = (FileUploadForm)form;

        // Process the FormFile
        FormFile file = myForm.getFile();
        String contentType = file.getContentType();
    //Get the file name
        String fileName    = file.getFileName();
       int fileSize       = file.getFileSize();
        byte[] fileData    = file.getFileData();
    //Get the servers upload directory real path name
    String filePath = getServlet().getServletContext().getRealPath("/") +"uploadfile";
    /* Save file on the server */
    if(!fileName.equals("")){
        System.out.println("Server path:" +filePath);
        //Create file
        File fileToCreate = new File(file, fileName);
        //If file does not exists create file
        if(!fileToCreate.exists()){
```

```
        FileOutputStream fileOutStream = new FileOutputStream(fileToCreate);
        fileOutStream.write(file.getFileData());
        fileOutStream.flush();
        fileOutStream.close();
      }


   }

      return mapping.findForward("success");
 }
```

File will be oploaded to "uploadfile" directory og your server.

Question No : 7

Q. What is DynaActionForm ? and How you can retrive the value which is set in the JSP Page in case of DynaActionForm ?

Answer :

DynaActionForm is specialized subclass of ActionForm that allows the creation of form beans with dynamic sets of properties,
without requiring the developer to create a Java class for each type of form bean.
DynaActionForm eliminates the need of FormBean class and now the form bean definition can be written into the struts-config.xml file. So, i
t makes the FormBean declarative and this helps the programmer to reduce the development time.

For Example : you have a EmpForm and you don't want a java class (EmpForm).
EmpForm has propertis
firstName, lastName, country

In the struts-config.xml file , declare the form bean

```
<form-bean name="EmpForm"
        type="org.apache.struts.action.DynaActionForm">
        <form-property name="firstName" type="java.lang.String"/>
        <form-property name="lastName" type="java.lang.String"/>
        <form-property name="country" type="java.lang.String" />
</form-bean>
```

Add action mapping in the struts-config.xml file:

```
<action path="/saveEmp" type="com.techfaq.action.EmpSaveAction"
```

```
    name="EmpForm"
    scope="request"
    validate="true"
    input="/pages/empform.jsp">

   <forward name="success" path="/jsp/success.jsp"/>
   <forward name="failure" path="/jsp/error.jsp" />

</action>
```

In the Action class.

```
public class EmpSaveAction extends Action
{
  public ActionForward execute(
    ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception{

    DynaActionForm empForm = (DynaActionForm)form;
    // this is the way you can retrive the value which is set in the JSP Page
    String firstName = (String)empForm.get("firstName");
    String lastName = (String)empForm.get("lastName");
    return mapping.findForward("success");
    }
  }
}
```

In the JSP page

```
<html:text property="firstName" size="30" maxlength="30"/>
<html:text property="lastName" size="30" maxlength="30"/>
```

Question No : 8

Q. How to Setup validator framework in Struts ?

Answer :

Step 1.

place validator-rules.xml and validation.xml in the WEB-INF directory.

Step 2.

Add the blow lines to struts-config.xml

```
 <plug-in className="org.apache.struts.validator.ValidatorPlugIn">
```

```
    <set-property property="pathnames" value="/WEB-INF/validator-rules.xml,
                                        /WEB-INF/validation.xml"/>
  </plug-in>
```

Step 3.
In the jsp add the tag
<html:errors />

Step 4.
Add
struts.jar
commons-beanutils.jar
commons-collections.jar
commons-digester.jar
commons-fileupload.jar
commons-lang.jar
commons-logging.jar
commons-validator.jar

into WEB-INF/lib directory

Explanation :

There are two configuration files used.
One if called validator-rules.xml and the other is validation.xml.
This is example of server side validation :

validator-rules.xml :

The validator-rules.xml file defines the Validator definitions available for a given application. The validator-rules.xml file acts as a template, defining all of the possible Validators that are available to an application.
Example validator-rules.xml File :

```
<form-validation>
 <global>
  <validator
    name="required"
    classname="org.apache.struts.util.StrutsValidator"
    method="validateRequired"
    methodparams="java.lang.Object,
                  org.apache.commons.validator.ValidatorAction,
```

```
                org.apache.commons.validator.Field,
                org.apache.struts.action.ActionErrors,
                javax.servlet.http.HttpServletRequest"
    msg="errors.required"/>

  </global>
</form-validation>
```

validation.xml File :
The validation.xml file is where you couple the individual Validators defined in the validator-rules.xml to components within your application.
validation.xml File

```
<form-validation>
 <formset>
  <form name="empForm">
    <field
      property="firstName"
      depends="required">
      <arg0 key="label.firstName"/>
    </field>

    <field
      property="lastName"
      depends="required">
      <arg0 key="label.lastName"/>
    </field>
  </form>
 </formset>
</form-validation>
```

In the empForm firstName and lastName are the required filed.
So in the above configuration you can see we add for both firstName and lastName.
You can see depends="required" - "required" property is defind in validator-rules.xml.

In the resource bundle : application_resource.propertis file

```
label.firstName=First Name
label.lastName=Last Name

#Error messages used by the Validator
errors.required={0} is required.
```

Based on the validation.xml File configuration .
Error in jsp will be :
First Name is required.
Last Name is required.

{0} will be filled by (First Name or Last Name) because validation.xml above configuration you have defind <arg0 key="label.firstName"/>.

Question No : 9
Q.How does validator framework work in Struts ?
Answer :
There are two configuration files used.
One if called validator-rules.xml and the other is validation.xml.
This is example of server side validation :

validator-rules.xml :

The validator-rules.xml file defines the Validator definitions available for a given application. The validator-rules.xml file acts as a template, defining all of the possible Validators that are available to an application.
Example validator-rules.xml File :

```
<form-validation>
 <global>
  <validator
    name="required"
    classname="org.apache.struts.util.StrutsValidator"
    method="validateRequired"
    methodparams="java.lang.Object,
                  org.apache.commons.validator.ValidatorAction,
                  org.apache.commons.validator.Field,
                  org.apache.struts.action.ActionErrors,
                  javax.servlet.http.HttpServletRequest"
    msg="errors.required"/>

  </global>
</form-validation>
```

validation.xml File :
The validation.xml file is where you couple the individual Validators defined in the validator-rules.xml to components within your application.

validation.xml File

```
<form-validation>
 <formset>
  <form name="empForm">
    <field
      property="firstName"
      depends="required">
      <arg0 key="label.firstName"/>
    </field>

    <field
      property="lastName"
      depends="required">
      <arg0 key="label.lastName"/>
    </field>
  </form>
 </formset>
</form-validation>
```

In the empForm firstName and lastName are the required filed.
So in the above configuration you can see we add for both firstName and lastName.
You can see depends="required" - "required" property is defind in validator-rules.xml.

In the resource bundle : application_resource.propertis file
label.firstName=First Name
label.lastName=Last Name

#Error messages used by the Validator
errors.required={0} is required.

Based on the validation.xml File configuration .
Error in jsp will be :
First Name is required.
Last Name is required.

{0} will be filled by (First Name or Last Name) because validation.xml above configuration you have defind <arg0 key="label.firstName"/>.

Question No : 10
What are Validators? and What are Basic Validators provided by the framework ?

Answer :
Validator is a Java class that, when called by the Validator framework, executes a validation rule. The framework knows how to invoke a Validator class based on its method signature, as defined in a configuration file.

1) byte,short,integer,long,float,double

2) creditCard - Checks if the field is a valid credit card number.
3) date - Checks if the field is a valid date.
4) email - Checks if the field is a valid email address.
4) mask - Succeeds if the field matches the corresponding regular expression mask.
5) maxLength - Checks if the value's length is less than or equal to the given maximum length.
6) minLength - Checks if the value's length is greater than or equal to the given minimum length.
7) range - Checks if the value is within a minimum and maximum range.
8) required - Checks if the field isn't null and length of the field is greater than zero, not including whitespace.

Question No : 11
What is the Benefits of Using the Validator framework in struts ?
Answer :
A few of the benefits include:

1) A single definition of validation rules for an application.

2) Validation rules are loosely coupled to the application.

3) Server-side and client-side validation rules can be defined in one location.

4) Configuration of new rules and/or changes to existing rules are made simpler.

5) Supports Internationalization.

6) Supports regular expressions.

7) Can be used for both Web-based and standard Java applications.

8) Promotes a declarative approach rather than a programmatic one.

Question No : 12
Q. What is the Difference between DispatchAction and LookupDispatchAction ?

Answer :
LookupDispatchAction is subclass of DispatchAction.
In case of DispatchAction, you have to declare the method name in the JSP page.
For Example :
http://localhost:8080/emp/empaction.do?step=add // IN CASE OF DispatchAction
here step=add , we are delaring method name in JSP.
or
<html:submit property="step">Add</html:submit> //IN CASE OF DispatchAction
so we can't use localization for button.

To over come both the issues below
a)method name declaration in JSP
b)can't use localization for button
We will go for LookupDispatchAction.

In the LookupDispatchAction :
For example :
If there are three operation like add, delete, update employee.
You can create three different Actions ?
AddEmpAction, DeleteEmpAction and UpdateEmpAction.
This is a valid approach, although not elegant since there might be duplication of code across the Actions since they are related. LookupDispatchAction is the answer to this problem. With LookupDispatchAction, you can combine all three Actions into one.

Example :
Step 1.
three buttons might be
<html:submit property="step">
<bean:message key="button.add"/>
</html:submit>
<html:submit property="step">
<bean:message key="button.delete"/>
</html:submit>
<html:submit property="step">
<bean:message key="button.update"/>
</html:submit>

//No need method name declaration in JSP . Button name from resource bundle.

Step 2.
In the the Resource Bundle. //Here you can add localization.

```
button.add=Add
button.delete=Delete
button.update=Update
```

Step 3.
Implement a method named getKeyMethodMap() in the subclass of the LookupDispatchAction. The method returns a java.util.Map. The keys used in the Map should be also used as keys in Message Resource Bundle.

```
public class EmpAction extends LookupDispatchAction {

public Map getKeyMethodMap()
{
Map map = new HashMap();
map.put("button.add", "add");
map.put("button.delete", "delete");
map.put("button.update", "update");
}


public ActionForward add(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("add-success");
}

public ActionForward delete(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("delete-success");
}

public ActionForward update(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
```

```
mapping.findForward("update-success");
}

}
```

in struts-config.xml

```
<action path="/empaction"
input="/empform.jsp"
type="list.EmpAction"
parameter="step"
scope="request"
validate="false">
<forward name="add-success"
path="addEmpSuccess.jsp"
redirect="true"/>
<forward name="delete-success"
path="deleteEmpSuccess.jsp"
redirect="true"/>
<forward name="update-success"
path="updateEmpSuccess.jsp"
redirect="true"/>
</action>
```

for every form submission, LookupDispatchAction does the
reverse lookup on the resource bundle to get the key and then gets the method
whose name is associated with the key from
getKeyMethodmap().

Flow of LookupDispatchAction:
a) Form the button name "Add" get the key "button.add" fro resource bundle.
b) then in the Map getKeyMethodMap() find the value associated with the key "button.add".
c) value from the Map is "add" . then call add() method of the same action class.

Question No : 13
Q. What is LookupDispatchAction?
Answer :
When a set of actions is closely related and
separating them into multiple Actions would result in duplication of code you can use
LookupDispatchAction.

LookupDispatchAction is subclass of DispatchAction.

In case of DispatchAction, you have to declare the method name in the JSP page.
For Example :
http://localhost:8080/emp/empaction.do?step=add // IN CASE OF DispatchAction
here step=add , we are delaring method name in JSP.
or
<html:submit property="step">Add</html:submit> //IN CASE OF DispatchAction
so we can't use localization for button.

To over come both the issues below
a)method name declaration in JSP
b)can't use localization for button
We will go for LookupDispatchAction.

For example :
If there are three operation like add, delete, update employee.
You can create three different Actions ?
AddEmpAction, DeleteEmpAction and UpdateEmpAction.
This is a valid approach, although not elegant since there might be duplication of code across the Actions since they are related. LookupDispatchAction is the answer to this problem. With LookupDispatchAction, you can combine all three Actions into one.

Example :
Step 1.
three buttons might be
<html:submit property="step">
<bean:message key="button.add"/>
</html:submit>
<html:submit property="step">
<bean:message key="button.delete"/>
</html:submit>
<html:submit property="step">
<bean:message key="button.update"/>
</html:submit>

//No need method name declaration in JSP . Button name from resource bundle.

Step 2.
In the the Resource Bundle. //Here you can add localization.
button.add=Add

button.delete=Delete
button.update=Update

Step 3.
Implement a method named getKeyMethodMap() in the subclass of the LookupDispatchAction. The method returns a java.util.Map. The keys used in the Map should be also used as keys in Message Resource Bundle.

```
public class EmpAction extends LookupDispatchAction {

public Map getKeyMethodMap()
{
Map map = new HashMap();
map.put("button.add", "add");
map.put("button.delete", "delete");
map.put("button.update", "update");
}


public ActionForward add(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("add-success");
}

public ActionForward delete(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("delete-success");
}

public ActionForward update(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("update-success");
```

}

}

in struts-config.xml

<action path="/empaction"
input="/empform.jsp"
type="list.EmpAction"
parameter="step"
scope="request"
validate="false">
<forward name="add-success"
path="addEmpSuccess.jsp"
redirect="true"/>
<forward name="delete-success"
path="deleteEmpSuccess.jsp"
redirect="true"/>
<forward name="update-success"
path="updateEmpSuccess.jsp"
redirect="true"/>
</action>

for every form submission, LookupDispatchAction does the
reverse lookup on the resource bundle to get the key and then gets the method
whose name is associated with the key from
getKeyMethodmap().

Flow of LookupDispatchAction:
a) Form the button name "Add" get the key "button.add" fro resource bundle.
b) then in the Map getKeyMethodMap() find the value associated with the key "button.add".
c) value from the Map is "add" . then call add() method of the same action class.

Question No : 14
Q.How to create a multiple selections list in Struts? and retrive seleted values ?
Answer :
This is the code to display multiple selections list and retrive seleted values in struts.
medList list contains list of Medium objects.

Java :
bean class.
public class Medium {

```
 int medId;
 String medName;

 /**
  * @return Returns the medId.
  */
 public int getMedId() {
  return medId;
 }
 /**
  * @param medId The medId to set.
  */
 public void setMedId(int medId) {
  this.medId = medId;
 }
 /**
  * @return Returns the medName.
  */
 public String getMedName() {
  return medName;
 }
 /**
  * @param medName The medName to set.
  */
 public void setMedName(String medName) {
  this.medName = medName;
 }
}
```

In the Form Class :

```
public class MediumForm {
private List medList;
private String[] med;
 public void setMedList(List medList){
  this.medList = medList;
 }

 public List getMedList(){
  return this.medList;
 }
```

```
 public String[] getMed() {
  return med;
 }

 public void setMed(String[] med) {
  this.med = med;
 }


}
```

In the Action class :

```
List medList = DAO.getMediums();
form.setMedList(medList);
```

DAO Class :

DAO Class to retrive mediums in data base.

```
public static List  getMediums(){
  PreparedStatement pStmt = null;
     Connection conn = null;
     boolean success = false;
     ResultSet rs = null;
    List medList = new ArrayList();

     try{
      conn =  getConnection();

      String sql = " select * from MEDIUM ";
      pStmt = conn.prepareStatement(sql);

      rs = pStmt.executeQuery();
      while(rs.next()){
       Medium med = new Medium();
       med.setMedId(rs.getInt("MED_ID"));
       med.setMedName(rs.getString("MEDIUM_NAME"));
       medList.add(med);
     }
```

```
    }catch(Exception e){
     e.printStackTrace();

    }finally{
     closeConnectionProp(conn,pStmt,rs);
    }

    return medList;

 }
```

JSP Struts:

```
 <html:select name="MediumForm" property="med"  multiple="true">

          <bean:define name="MediumForm" property="medList" id="mlist" />
           <html:options collection="mlist" property="medId"  labelProperty="medName" />
</html:select>
```

In the Action class again :
how to retrive the selected values.

```
String[] med = form.getMed();
```

Question No : 15
Q.How to create a drop down list in Struts?
Answer :
This is the code to display drop down list and retrive seleted value in struts.
medList list contains list of Medium objects.

Java :
bean class.

```
public class Medium {
 int medId;
 String medName;

 /**
  * @return Returns the medId.
```

```
 */
 public int getMedId() {
  return medId;
 }
 /**
  * @param medId The medId to set.
  */
 public void setMedId(int medId) {
  this.medId = medId;
 }
 /**
  * @return Returns the medName.
  */
 public String getMedName() {
  return medName;
 }
 /**
  * @param medName The medName to set.
  */
 public void setMedName(String medName) {
  this.medName = medName;
 }
}
```

In the Form Class :

```
public class MediumForm {
private List medList;
private String med;
 public void setMedList(List medList){
  this.medList = medList;
 }

 public List getMedList(){
  return this.medList;
 }

 public void setMed(String medList){
  this.med = med;
 }

 public String getMed(){
```

```
  return this.med;
 }


}
```

In the Action class :

```
List medList = DAO.getMediums();
form.setMedList(medList);
```

DAO Class :

DAO Class to retrive mediums in data base.

```
public static List  getMediums(){
  PreparedStatement pStmt = null;
     Connection conn = null;
     boolean success = false;
     ResultSet rs = null;
    List medList = new ArrayList();

     try{
      conn =  getConnection();

      String sql = " select * from MEDIUM ";
      pStmt = conn.prepareStatement(sql);

      rs = pStmt.executeQuery();
      while(rs.next()){
       Medium med = new Medium();
       med.setMedId(rs.getInt("MED_ID"));
       med.setMedName(rs.getString("MEDIUM_NAME"));
       medList.add(med);
     }



     }catch(Exception e){
      e.printStackTrace();

     }finally{
      closeConnectionProp(conn,pStmt,rs);
```

```
    }

    return medList;

 }
```

JSP Struts:

```
 <html:select name="MediumForm" property="med" >

          <bean:define name="MediumForm" property="medList" id="mlist" />
           <html:options collection="mlist" property="id"  labelProperty="medName" />
</html:select>
```

In the Action class again :
how to retrive the select value.

```
String med = form.getMed();
```

Question No : 16
Q. What is DispatchAction ?
Answer :
When a set of actions is closely related and
separating them into multiple Actions would result in duplication of code you can use DispatchAction.
For example :
If there are three operation like add, delete, update employee.
You can create three different Actions ?
AddEmpAction, DeleteEmpAction and UpdateEmpAction.
This is a valid approach, although not elegant since there might be duplication of code across the Actions since they are related. DispatchAction is the answer to this
problem. With DispatchAction, you can combine all three Actions into one.

three urls might be
http://localhost:8080/emp/empaction.do?step=add
http://localhost:8080/emp/empaction.do?step=delete
http://localhost:8080/emp/empaction.do?step=update

```
public class EmpAction extends DispatchAction {
public ActionForward add(ActionMapping mapping,
```

```
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("add-success");
}

public ActionForward delete(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("delete-success");
}

public ActionForward update(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("update-success");
}

}
```

in struts-config.xml

```
<action path="/empaction"
input="/empform.jsp"
type="list.EmpAction"
parameter="step"
scope="request"
validate="false">
<forward name="add-success"
path="addEmpSuccess.jsp"
redirect="true"/>
<forward name="delete-success"
path="deleteEmpSuccess.jsp"
redirect="true"/>
<forward name="update-success"
path="updateEmpSuccess.jsp"
redirect="true"/>
```

</action>

Notice that one of the HTTP request parameter in the four URLs is also named "step". Now, it is all coming
together. DispatchAction knows what parameter to look for in the incoming
URL request through this attribute named parameter in struts-config.xml.

Question No : 17
Q. What is IncludeAction ?
Answer :
IncludeAction included resulting resource in the HTTP response .
In the JSP which don't use struts for include we do
<jsp:include page="/test/ServletA"/>

In struts JSP
<jsp:include page="/App1/legacyA.do" />

In the struts-config.xml
<action path="/legacyA"
parameter="/test/LegacyServletA"
type="org.apache.struts.actions.IncludeAction" />

Question No : 18
Q. How to Protect JSPs from direct access ?
Answer :
JSPs located in the WEB-INF and its sub-directories are protected from outside access.
  If you want to go pageB.jsp from pageA.jsp
  <html:link action="gotoPageB">Go to Page B</html:link>

  in the struts-config.xml
  <action path="/gotoPageB"
parameter="/WEB-INF/pageB.jsp"
type="org.apache.struts.actions.ForwardAction"/>

Question No : 19
Q. What is ForwardAction ?
Answer :
Suppose you want to go from PageA.jsp to PageB.jsp in your Struts
application. The easy way of achieving this is to add a hyperlink in PageA.jsp as
follows:
Go to Page B
or even better, as follows:

<html:link page="/PageB.jsp">Go to Page B</html:link>
However this violates the MVC spirit by directly accessing
the JSP.
Struts provides a built-in Action class called ForwardAction to address this
issue. With ForwardAction, the Struts Controller is still in the loop while
navigating from PageA to PageB. There are two steps involved in using the
ForwardAction. They are:
_ First, declare the PageA hyperlink that takes you to PageB as follows:
<html:link page="/gotoPageB.do">Go to Page B</html:link>
_ Next, add an ActionMapping in the Struts Config file as follows:
<action path="/gotoPageB"
parameter="/PageB.jsp"
type="org.apache.struts.actions.ForwardAction" />

Question No : 20
How does reset() and Validate() method struts work ?
Answer :
reset(): reset() method is called by Struts Framework with each request that uses the defined ActionForm. The purpose of this method is to reset all of the ActionForm's data members prior to the new request values being set.
Example:

```
/**
   * Reset the form.
   */
  public void reset(ActionMapping mapping, HttpServletRequest request) {

   super.reset(mapping,request);
      this.password = null;
      this.username = null;
      this.guest = null;
  }
```

validate() : Used to validate properties after they have been populated; Called before FormBean is handed to Action. Returns a collection of ActionError as ActionErrors. Following is the method signature for the validate() method.

Example :

```
/**
   * Validate the form's input.
   */
  public ActionErrors validate(ActionMapping mapping,
    HttpServletRequest request) {
```

```
   ActionErrors errors = new ActionErrors();

if(username == null){
     errors.add(ActionErrors.GLOBAL_ERROR, new ActionError("error.username.required"));
    }
return errors;
}
```

Question No : 21

Multiple buttons in struts using java script?

Answer :

Example :

JSP Page

```
<html:form action="/organization.do?method=createPage" >
<tr>
   <td>Fax</td>
   <td><html:text property="fax" size="20"/></td>
  </tr>
  <tr>
   <td>Email</td>
   <td><html:text property="email" size="20"/></td>
  </tr>

<input type="button" value="Save and Continue" name="B1" onClick="save()">
   <input type="button" value="Discard Changes" name="B2" onClick="discard()">
</html:form>
```

In the Java Script

```
<script type="text/javascript" >

 function save() {

  document.OrganizationForm.action
="<%=request.getContextPath()%>/organization.do?method=save";
  document.OrganizationForm.submit();
 }

 function discard() {

  document.OrganizationForm.action
```

="<%=request.getContextPath()%>/organization.do?method=discard";
  document.OrganizationForm.submit();
 }
</script>

Question No : 22
Integration Struts Spring Hibernate ?
Answer :
Code is here with explanation

[URL=http://www.techfaq360.com/tutorial/spring/struts_spring_hibernate.jsp]http://www.techfaq360.com/tutorial/spring/struts_spring_hibernate.jsp[/URL]

Step 1.
In the struts-config.xml add plugin
<plug-in className="org.springframework.web.struts.ContextLoaderPlugIn">
        <set-property property="contextConfigLocation"
            value="/WEB-INF/applicationContext.xml"/>
    </plug-in>

Step 2.

In the applicationContext.xml file
Configure datasourse
    <bean id="dataSource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
     <property name="driverClassName"><value>oracle.jdbc.driver.OracleDriver</value>
      </property>
        <property name="url"><value>jdbc:oracle:thin:@10.10.01.24:1541:ebizd</value>
        </property>
        <property name="username"><value>sa</value></property>
   <property name="password"><value></value></property>
    </bean>

Step 3.

Configure SessionFactory
 <!-- Hibernate SessionFactory -->
    <bean id="sessionFactory"
class="org.springframework.orm.hibernate.LocalSessionFactoryBean">

```
        <property name="dataSource"><ref local="dataSource"/></property>
        <property name="mappingResources">
           <list>
              <value>com/test/dbxml/User.hbm.xml</value>
           </list>
        </property>
        <property name="hibernateProperties">
        <props>
           <prop key="hibernate.dialect">net.sf.hibernate.dialect.OracleDialect </prop>
         </props>
        </property>
    </bean>
```

Step 4.
Configure User.hbm.xml

```
<hibernate-mapping>
    <class name="org.test.model.User" table="app_user">

        <id name="id" column="id" >
           <generator class="increment"/>
        </id>
        <property name="firstName" column="first_name" not-null="true"/>
        <property name="lastName" column="last_name" not-null="true"/>

    </class>
</hibernate-mapping>
```

Step 5.

In the applicationContext.xml ? configure for DAO

```
 <bean id="userDAO" class="org.test.dao.hibernate.UserDAOHibernate">
        <property name="sessionFactory"><ref local="sessionFactory"/></property>
    </bean>
```

Step 6.

DAO Class

```
public class UserDAOHibernate extends HibernateDaoSupport implements UserDAO {
    private static Log log = LogFactory.getLog(UserDAOHibernate.class);
```

```
    public List getUsers() {
        return getHibernateTemplate().find("from User");
    }

    public User getUser(Long id) {
        return (User) getHibernateTemplate().get(User.class, id);
    }

    public void saveUser(User user) {
        getHibernateTemplate().saveOrUpdate(user);

        if (log.isDebugEnabled()) {
            log.debug("userId set to: " + user.getId());
        }
    }

    public void removeUser(Long id) {
        Object user = getHibernateTemplate().load(User.class, id);
        getHibernateTemplate().delete(user);
    }
}
```

Question No : 23

Q.Mutli-click prevention using struts tokens with code example.

Answer :

Struts has 3 methods use for the token, saveToken(), isTokenValid() and resetToken().

saveToken() - generate the token key and save to request/session attribute.

isTokenValid() - validate submitted token key against the 1 store in request/session.

resetToken() - reset the token key

Example :

Step 1.

Action Class where saveToken() before JSP Page.

First saveToken() then forward to your jsp.

Upon loading the form, invokes saveToken() on the action class to create and store the token key.

Struts will store the generated key in request/session.

```
public class LoadAction extends Action
{
 public ActionForward execute(ActionMapping  mapping,ActionForm form,HttpServletRequest
```

```
request,HttpServletResponse response)
 { ActionForward forward;
  forward=mapping.findForward("FirstPage");// this is the jsp page where you want to struts tokens.
  saveToken(request);

  return forward;
 }
}
```

Step 2.
If the token successfully created, when view source on the browser you will see the token, the token key is stored as a hidden field:
In jsp page :

```
<%@ page import="org.apache.struts.action.Action"%>
<%@ page import="org.apache.struts.taglib.html.Constants"%>

<%@ taglib uri="/WEB-INF/struts-tiles.tld" prefix="tiles" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<html>
<head> <title> First Page </title> </head>

<body>
<form name="MyForm" method="post" action="/dpsubm/getForm/submit.do">
<input type="text" name="name" >
 <input type="hidden"   name="<%= Constants.TOKEN_KEY %>"
        value="<%= session.getAttribute(Action.TRANSACTION_TOKEN_KEY) %>" >
<input type="submit"  value="submit">
</form>
</body>
</html>
```

Step 3. Your logic
Once the form submitted, invokes isTokenValid() on the action class, it will validate the submitted token key(hidden field) with the token key stored previously on request/session. If match, it will return true.

```
public class SubmitAction extends Action
{
 public ActionForward execute(ActionMapping mapping ,ActionForm form ,HttpServletRequest
```

```
request,HttpServletResponse response)
 {
  ActionForward forward=mapping.findForward("submitForm");
  DupSubmitForm frm=(DupSubmitForm)form;

  if(isTokenValid(request))
  {

   System.out.println("frm.getName()"+frm.getName());
   resetToken(request);
  }
  else
  {
   System.out.println("frm.getName()"+frm.getName());
   System.out.println("Duplicate Submission of the form");

  }
  return forward;
 }
}
```

Code you can get from:
http://www.techfaq360.com/viewTutorial.jsp?tutorialId=62

Question No : 24

Q. How to prevent mutli-click using struts tokens ?

Answer :

Struts has 3 methods use for the token, saveToken(), isTokenValid() and resetToken().

saveToken() - generate the token key and save to request/session attribute.
isTokenValid() - validate submitted token key against the 1 store in request/session.
resetToken() - reset the token key.

How it works:

1. Upon loading the form, invokes saveToken() on the action class to create and store the token key. Struts will store the generated key in request/session. If the token successfully created, when view source on the browser you will see something similar to the following, the token key is stored as a hidden field:

```
<form action="myaction.do" method="post">
 <input type="hidden"
        name="<%= Constants.TOKEN_KEY %>"
        value="<%= session.getAttribute(Action.TRANSACTION_TOKEN_KEY) %>" >
```

2. Once the form submitted, invokes isTokenValid() on the action class, it will validate the submitted token key(hidden field) with the token key stored previously on request/session. If match, it will return true.

.
In Action Class.

```
public final ActionForward perform(ActionMapping mapping,
                                   ActionForm form,
                                   HttpServletRequest request,
                                   HttpServletResponse response)
          throws IOException, ServletException
    {
      saveToken(request);
      if (!tokenIsValid(request))
      {
            //forward to error page saying "your transaction is already being  processed"
      else
      {
            //process action
            //forward to jsp
      }

       // Reset token after transaction success.
       resetToken(request);

 }
```

Details you can check
[URL=http://www.techfaq360.com/tutorial/multiclick.jsp]http://www.techfaq360.com/tutorial/multiclick.jsp[/URL]

Question No : 25
Q.How you will enable front-end validation based on the xml in validation.xml?
Answer :
The <html:javascript> tag to allow front-end validation based on the xml in validation.xml. For example the code: <html:javascript formName=?logonForm? dynamicJavascript=?true? staticJavascript=?true? /> generates the client side java script for the form ?logonForm? as defined in the validation.xml file. The <html:javascript> when added in the jsp file generates the client site validation script.

Question No : 26
Q.What is new in ServletRequest interface ?
Answer :
The following methods have been added to ServletRequest 2.4 version:
public int getRemotePort()
public java.lang.String getLocalName()
public java.lang.String getLocalAddr()
public int getLocalPort()

Question No : 27
Q.Struts Action Chaining?
Answer :
Chaining actions can be done by simply using the
proper mapping in your forward entries in the struts-config.xml file.

```
public class AAction extends Action
{
public ActionForward
execute(ActionMapping mapping,
ActionForm form,
HttpServletRequest request,
HttpServletResponse response) throws
Exception
{
// Do something

return mapping.findForward("success");
}
}
```

```
/* com/BAction.java */
...

public class BAction extends Action
{
public ActionForward
execute(ActionMapping mapping,
ActionForm form,
HttpServletRequest request,
```

HttpServletResponse response) throws
Exception
{
// Do something else

return mapping.findForward("success");
}
}

Then you can chain together these two actions with
the Struts configuration as shown in the following excerpt:

...
<action-mappings type="org.apache.struts.action.ActionMapping">
<action path="/A"
type="com.AAction"
validate="false">
<forward name="success" path="/B.do" />
</action>
<action path="/B"
type="com.BAction"
scope="session"
validate="false">
<forward name="success" path="/result.jsp" />
</action>
</action-mappings>

Question No : 28
Q.How can I avoid validating a form before data is entered?

Answer :
validate="false" ..
The simplest way is to have two actions. The first one has the job of setting the form data, i.e. a blank registration screen. The second action in our writes the registration data to the database. Struts would take care of invoking the validation and returning the user to the correct screen if validation was not complete.
The EditRegistration action in the struts example application illustrates this:
< action path="/editRegistration">
type="org.apache.struts.webapp.example.EditRegistrationAction"

attribute="registrationForm"
scope="request"
validate="false">
<forward name="success path="/registration.jsp"/>
</action>
When the /editRegistration action is invoked, a registrationForm is created and added to the request, but its validate method is not called. The default value of the validate attribute is true, so if you do not want an action to trigger form validation, you need to remember to add this attribute and set it to false
---
Question No : 29
Q.Can I have an Action without a form?
Answer :
Yes. If your Action does not need any data and it does
not need to make any data available to the view or
controller component that it forwards to, it doesn't need
a form. A good example of an Action with no ActionForm is
the LogoffAction in the struts example application:

<action path="/logoff"
type="org.apache.struts.webapp.example.LogoffAction">
<forward name="success" path="/index.jsp"/>
</action>

This action needs no data other than the user's session, which
it can get from the Request, and it doesn't need to prepare any
view elements for display, so it does not need a form.

However, you cannot use the <html:form> tag without
an ActionForm. Even if you want to use the <html:form>
tag with a simple Action that does not require input,
the tag will expect you to use some type of ActionForm,
even if it is an empty subclass without any properties.

Question No : 30
Q.What is Struts Validator Framework?
Answer :
Struts Framework provides the functionality to validate the form data. It can be use to validate the data on the users browser as well as on the server side. Struts Framework emits the java scripts and it can be used validate the form data on the client browser. Server side validation of form can

be accomplished by sub classing your From Bean with DynaValidatorForm class.

Client side validation : validation-rule.xml (java script code should be in this file and in JSP file <html:javascript formName="CreateForm"/>)

Server side Validation :
JSP page - should cointain
< html:errors/ >
Aactionform.validate() method
or
The Validator Framework uses two XML configuration files validator-rules.xml and validation.xml. The validator-rules.xml defines the standard validation routines, these are reusable and used in validation.xml. to define the form specific validations. The validation.xml defines the validations applied to a form bean.
How you will display validation fail errors on jsp page? -
The following tag displays all the errors: < html:errors/ >

Get more details on
[URL=http://www.techfaq360.com/tutorial/validation_framework.jsp]http://www.techfaq360.com/tutorial/validation_framework.jsp[/URL]

Question No : 31
Q.How you will make available any Message Resources Definitions file to the Struts Framework Environment?

Answer :
Message Resources Definitions file are simple .properties files and these files contains the messages that can be used in the struts project. Message Resources Definitions files can be added to the struts-config.xml file through < message-resources / > tag. Example: < message-resources parameter= MessageResources / >

Question No : 32
Q.What helpers in the form of JSP pages are provided in Struts framework?
Answer :
--struts-html.tld
--struts-bean.tld
--struts-logic.tld

Question No : 33

Q.How you will enable front-end client side validation based on the xml in validation.xml?

Answer :
Validation Framework-Client Side
Validation Framework provides the functionality to validate the form data. It can be use to validate the data on the client side. Errors will be displayed like java script.

Struts has a class called ValidatorForm in org.apache.struts.validator package. This is a subclass of ActionForm and implements the validate() method. The validate() method invokes the Commons Validator, executes the rules using the two xml files (validator-rules.xml and validation.xml) and generates ActionErrors using the Message Resources defined in the struts-config.xml.
validator-rules.xml :
The validator-rules.xml file defines the Validator definitions available for a given application. The validator-rules.xml file acts as a template, defining all of the possible Validators that are available to an application.
validation.xml File :
The validation.xml file is where you couple the individual Validators defined in the validator-rules.xml to components within your application.

Follow the below steps to setup Validation Framework in Struts (server side validation).

Step 1. Add validator plugin into struts-config.xml

<plug-in className="org.apache.struts.validator.ValidatorPlugIn"> <set-property property="pathnames" value="/WEB-INF/validator-rules.xml, /WEB-INF/validation.xml"/>
</plug-in>

Step 3. In the JSP page (EmpForm.jsp)- Add onsubmit="return validateempForm(this);" and <html:javascript formName="empForm"/> for Client Side Java Script Validation.

<%@ taglib uri="/WEB-INF/struts-tiles.tld" prefix="tiles" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<html:form>
<html:form action="/submitForm.do" method="post" onsubmit="return validateempForm(this);">
<html:errors/> // this is for error message from resource bundle in JSP
<html:text property="firstName" size="20" maxlength="50"/>
<html:text property="lastName" size="20" maxlength="50"/>

```
<html:submit >Save</html:submit>
<!-- Begin Validator Javascript Function-->
<html:javascript formName="empForm"/>
<!-- End of Validator Javascript Function-->
</html:form>
```

Step 4. Add Message Resources location in struts-config.xml

<!-- Message Resources --> <message-resources parameter="application_resource"/>

Step 5. In the the Resource Bundle. application_resource.properties file //Here you can add localization.

```
label.firstName=First Name
label.lastName=Last Name
errors.required={0} is required.
```

Step 7. In the EmpForm

```
package com.techfaq.form;
public class EmpForm extends ValidatorForm {
int empId;
String firstName;
String lastName;
public String getFirstName() {
return firstName;
}
public void setFirstName(String firstName) {
this.firstName = firstName;
}
public String getLastName() {
return lastName;
}
public void setLastName(String lastName) {
this.lastName = lastName;
}
public int getEmpId() {
return empId;
}
public void setEmpId(int empId) {
```

```
this.empId = empId;
}
}
```

Step 6. In the validation.xml - The validation.xml file is where you couple the individual Validators defined in the validator-rules.xml to components within your application

```
<form-validation>
<formset>
<form name="empForm">
<field property="firstName" depends="required">
<arg0 key="label.firstName"/>
</field>
<field property="lastName" depends="required"> <arg0 key="label.lastName"/>
</field>
</form>
</formset>
</form-validation>
```

Step 6. In the validator-rules.xml - The validator-rules.xml file defines the Validator definitions available for a given application.

```
<form-validation>
<global>
<validator
name="required"
classname="org.apache.struts.util.StrutsValidator"
method="validateRequired"
methodparams="java.lang.Object,
org.apache.commons.validator.ValidatorAction,
org.apache.commons.validator.Field,
org.apache.struts.action.ActionErrors,
javax.servlet.http.HttpServletRequest"
msg="errors.required"/>
</global>
</form-validation>
```

Step 5. Add Action mapping and form entry into the stuts-confix.xml and validate="true" is for validation framework to validate the form input.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 1.1//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">
<struts-config>
<form-beans>
<form-bean name="EmpForm" type="com.techfaq.form.EmpForm"> </form-bean>
</form-beans>
<action-mappings>
<action path="/submitForm"
type="com.techfaq.action.EmpAction"
name="EmpForm"
scope="request"
validate="true" // This need for validation framework to validate the form input
input="EmpForm.jsp">
<forward name="success"
path="success.jsp"/>
<forward name="failure" path="failure.jsp" />
</action>
</action-mappings>
</struts-config>
```

Now in the browser type http://localhost:8080/testApp/EmpForm.jsp

Don't Enter firstName and lastName in the text box and submit the "Save" BUTTON. the RequestProcessor checks for the validateattribute in the ActionMapping.
If the validate is set to true, the RequestProcessor invokes the validate() method of the ValidatorForm instance.
If Validate fail the RequestProcessor looks for the input attribute and return to JSP page mentioned in input tag.
If Validate pass goto Action Class execute() method..
If Validate fail , In the browser (EmpForm.jsp) you can see. Java Script pop up Message:

First Name is required.
Last Name is required.

In the empForm firstName and lastName are the required filed. So in the above configuration you can see we add for both firstName and lastName. You can see depends="required" - "required" property is defind in validator-rules.xml. In the resource bundle : application_resource.propertis file

label.firstName=First Name
label.lastName=Last Name
#Error messages used by the Validator
errors.required={0} is required.
{0} will be filled by (First Name or Last Name) because validation.xml above configuration you have defind
<arg0 key="label.lastName"/>. and <arg0 key="label.lastName"/>.

Question No : 34
Q.What design patterns are used in Struts?
Answer :
Struts is based on model 2 MVC (Model-View-Controller) architecture. Struts controller uses the command design pattern and the action classes use the adapter design pattern. The process() method of the RequestProcessor uses the template method design pattern. Struts also implement the following J2EE design patterns.

Service to Worker
Dispatcher View
Composite View (Struts Tiles)
Front Controller
View Helper
Synchronizer Token

Question No : 35
Q.What is role of Action Class?
Answer :
An Action Class performs a role of an adapter between the contents of an incoming HTTP request and the corresponding business logic that should be executed to process this request.

In the execute() method of Action class the business logic is executed.

public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception ;

execute() method of Action class:

Perform the processing required to deal with this request

Update the server-side objects (Scope variables) that will be used to create the next page of the user interface
Return an appropriate ActionForward object

Question No : 36
Q.What is ActionMapping and is the Action Mapping specified?
Answer :
Action mapping contains all the deployment information for a particular Action bean. This class is to determine where the results of the Action will be sent once its processing is complete.
We can specify the action mapping in the configuration file called struts-config.xml. Struts framework creates ActionMapping object from <ActionMapping> configuration element of struts-config.xml file

<action-mappings>
<action path="/submit" type="com.SubmitAction" name="submitForm" input="/submit.jsp" scope="request" validate="true"> <forward name="success" path="/success.jsp"/>
<forward name="failure" path="/error.jsp"/> </action>
</action-mappings>

Question No : 37
Q.What is the ActionForm and what are important methods in ActionForm?
Answer :
ActionForm is javabean which represents the form inputs containing the request parameters from the View referencing the Action bean.

The important methods of ActionForm are : validate() & reset().

validate() : Used to validate properties after they have been populated; Called before FormBean is handed to Action. Returns a collection of ActionError as ActionErrors. Following is the method signature for the validate() method.

ActionErrors validate(ActionMapping mapping,HttpServletRequest request)

reset(): reset() method is called by Struts Framework with each request that uses the defined ActionForm. The purpose of this method is to reset all of the ActionForm's data members prior to the new request values being set.

Question No : 38
Q.What is role of ActionServlet?

Answer :
ActionServlet performs the role of Controller:

Process user requests
Determine what the user is trying to achieve according to the request
Pull data from the model (if necessary) to be given to the appropriate view,
Select the proper view to respond to the user
Delegates most of this grunt work to Action classes
Is responsible for initialization and clean-up of resources

Question No : 39
Q.What is ActionServlet?
Answer :
The class org.apache.struts.action.ActionServlet is the called the ActionServlet. In the the Jakarta Struts Framework this class plays the role of controller. All the requests to the server goes through the controller. Controller is responsible for handling all the requests.

Struts Flow start with ActionServlet then call to process() method of RequestProcessor.

Step 1. Load ActionServlet using load-on-startup and do the following tasks.

Any struts web application contain the ActionServlet configuration in web.xml file.

On load-on-startup the servlet container Instantiate the ActionServlet .

First Task by ActionServlet : The ActionServlet takes the Struts Config file name as an init-param.

At startup, in the init() method, the ActionServlet reads the Struts Config file and load into memory.

Second Task by ActionServlet : If the user types http://localhost:8080/app/submitForm.do in the browser URL bar, the URL will be intercepted and processed by the ActionServlet since the URL has a pattern *.do, with a suffix of "do". Because servlet-mapping is

<servlet-mapping>

<servlet-name>action</servlet-name>

<url-pattern>*.do</url-pattern>

</servlet-mapping>

Third Task by ActionServlet : Then ActionServlet delegates the request handling to another class called RequestProcessor by invoking its process() method.

Step 2. ActionServlet calls process() method of RequestProcessor

---

Question No : 40
What are the components of Struts?
Answer :
Struts components can be categorize into Model, View and Controller:

Model: Components like business logic /business processes and data are the part of model.
View: HTML, JSP are the view components.
Controller: Action Servlet of Struts is part of Controller components which works as front controller to handle all the requests.

Question No : 41
What is MVC and how it maps to Struts?
Answer :
Model-View-Controller (MVC) is a design pattern put together to help control change. MVC decouples interface from business logic and data.

Model : The model contains the core of the application's functionality. The model encapsulates the state of the application. Sometimes the only functionality it contains is state. It knows nothing about the view or controller.
Action class in struts.

View: The view provides the presentation of the model. It is the look of the application. The view can access the model getters, but it has no knowledge of the setters. In addition, it knows nothing about the controller. The view should be notified when changes to the model occur.
JSP in struts

Controller:The controller reacts to the user input. It creates and sets the model.

ActionServlet and RequestProcessor class in struts.

Question No : 42

How do you get a password field in struts ?

Answer :

<html:password property = "password" maxlength = "60" />

Question No : 43

Q.Struts Flow In Depth?

Answer :

Steps 1.

ActionServlet

The central component of the Struts Controller is the ActionServlet. It is a concrete class and extends the javax.servlet.HttpServlet. It performs two important things.

On startup, its reads the Struts Configuration file and loads it into memory in the init() method.

In the doGet() and doPost() methods, it intercepts HTTP request and handles it appropriately.

In the web.xml

```
<servlet>
<servlet-name>action</servlet-name>
<servlet-class>org.apache.struts.action.ActionServlet
</servlet-class>
<init-param>
<param-name>config</param-name>
<param-value>/WEB-INF/struts-config.xml</param-value>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
<servlet-name>action</servlet-name>
<url-pattern>*.do</url-pattern>
</servlet-mapping>
```

If the user types http://localhost:8080/App1/submitDetails.do in the browser URL bar. Server will call ActionServlet class because in the url-pattern the mapping is

<url-pattern>*.do</url-pattern>. Any *.do will call ActionServlet class.
ActionServlet calls the process() method of RequestProcessor class

Step 2.
ActionServlet calls the process() method of RequestProcessor class.
The RequestProcessor first retrieves appropriate XML block for
the URL from struts-config.xml. This XML block is referred to as
ActionMapping in Struts terminology. In fact there is a class called
ActionMapping in org.apache.struts.action package.
ActionMapping is the class that does what its name says ? it holds the mapping
between a URL and Action.

A sample ActionMapping from struts-config.xml

<action path="/submitDetails"
type="mybank.example.CustomerAction"
name="CustomerForm"
scope="request"
validate="true"
input="CustomerDetailForm.jsp">
<forward name="success"
path="ThankYou.jsp"
redirect=?true?/>
<forward name="failure" path="error.jsp" />
</action>

Step 3.

The RequestProcessor looks up the configuration file for the URL
pattern /submitDetails. and finds the XML block (ActionMapping) shown above.
The type attribute tells Struts which Action class has to be instantiated.

Step 4.
The RequestProcessor instantiates the CustomerForm and puts
it in appropriate scope ? either session or request. The RequestProcessor
determines the appropriate scope by looking at the scope attribute in the same
ActionMapping.

Step 5.
Next, RequestProcessor iterates through the HTTP request parameters
and populates the CustomerForm properties of the same name as the HTTP
request parameters using Java Introspection.

Step 6.
Next, the RequestProcessor checks for the validate attribute in the ActionMapping. If the validate is set to true, the RequestProcessor invokes the validate() method on the CustomerForm instance. This is the method where you can put all the html form data validations. If any error then RequestProcessor checks for the input attribute in the ActionMapping and forward to page mentioned in the input tag.
If no error in validate() method then continue.

Step 7.

The RequestProcessor instantiates the Action class specified in the ActionMapping (CustomerAction) and invokes the execute() method on the CustomerAction instance. The signature of the execute method is as follows.
public ActionForward execute(ActionMapping mapping,
ActionForm form,
HttpServletRequest request,
HttpServletResponse response) throws Exception

The execute() method returns ActionForward.

ActionForward forward = mapping.findForward(?success?);
return forward. will forward to ThankYou.jsp.
ActionForward forward = mapping.findForward(failure);
return forward. will forward to error.jsp.

Question No : 44
Q.How does validate() method of ActionForm work ?
Answer :
<action path="/submitDetails"
type="mybank.example.CustomerAction"
name="CustomerForm"
scope="request"
validate="true"
input="CustomerDetailForm.jsp">
<forward name="success"
path="ThankYou.jsp"
redirect=?true?/>
<forward name="failure" path="error.jsp" />
</action>

RequestProcessor checks for the validate attribute in the
ActionMapping. If the validate is set to true, the RequestProcessor invokes
the validate() method on the CustomerForm instance. This is the method
where you can put all the html form data validations. If any error then
RequestProcessor checks for the input attribute in the ActionMapping
and forward to page mentioned in the input tag.
If no error in validate() method then continue.

and Validate() method looks like

```
public ActionErrors validate(ActionMapping mapping,
HttpServletRequest request)
{
// Perform validator framework validations
ActionErrors errors = new ActionErrors();
// Only need crossfield validations here
if (parent == null) {
errors.add(ActionErrors.GLOBAL_ERROR,
new ActionError("error.custform"));
}
if (firstName == null) {
errors.add(ActionErrors.GLOBAL_ERROR,
new ActionError("error.firstName.null"));
}
return errors;
}
```

where error.custform and error.firstName.null keys from Message Resource Bundle.

Question No : 45

Q.What are the important sections in Struts Configuration File ? struts-config.xml?

Answer :

The five important sections are:

1. Form bean definition section
2. Global forward definition section
3. Action mapping definition section
4. Controller configuration section
5. Application Resources definition section

```
<struts-config>
<form-beans>
 <form-bean name="CustomerForm" type="example.CustomerForm"/>
```

```
 <form-bean name="LogonForm" type="example.LogonForm"/>
</form-beans>

<global-forwards>
 <forward name="logon" path="/logon.jsp"/>
 <forward name="logoff" path="/logoff.do"/>
</global-forwards>
<action-mappings>
 <action path="/submitDetailForm"
 type="example.CustomerAction"
 name="CustomerForm"
 scope="request"
 validate="true"
 input="/CustomerDetailForm.jsp">
 <forward name="success" path="/ThankYou.jsp" redirect=?true? />
 <forward name="failure" path="/Failure.jsp" />
 </action>

</action-mappings>
<controller processorClass="org.apache.struts.action.RequestProcessor" />
<message-resources parameter="example.resource.ApplicationResources"/>
```

Question No : 46
Q.How to handle Handling multiple buttons in HTML Form ?
Answer :
You have to use a variation of the <html:submit> as shown below to tackle this problem.

```
<html:submit property=?step?>
<bean:message key=?button.save?/>
</html:submit>

<html:submit property=?step?>
<bean:message key=?button.delete"/>
</html:submit>
```

In the Message Resource

```
button.save = Save Me;
button.delete = Delete
```

you have to add a variable named ?step? in the ActionForm and then add

two methods getStep() and setStep()

In the execute() method.
public ActionForward execute(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{

CustomerForm custForm = (CustomerForm) form;
ActionForward forward = null;
if (?Save Me?.equals(custForm.getStep())) {
System.out.println(?Save Me Button Clicked?);
} else if(?Delete?.equals(custForm.getStep())){
System.out.println(?Delete Button Clicked?);
}
return forward;
}

Question No : 47
Q.How does Value replacement in Message Resource Bundle work?
Answer :
In the resource bundle file, you can define a template like:
errors.required={0} is required.
ActionErrors errors = new ActionErrors();
errors.add(ActionErrors.GLOBAL_ERROR,
new ActionError("error.custform","First Name"));

Then the Error message is : First Name is required.

Other constructors are

public ActionError(String key, Object value0, Object value1)
. . .
public ActionError(String key, Object[] values);

Question No : 48
What is SwitchAction?
Answer :
The SwitchAction class provides a means to switch from a resource in one module to another resource in a different module. SwitchAction is useful only if you have multiple modules in your Struts application and multiple struts-config.xml. The SwitchAction class can be used as is, without extending.

Question No : 49

What is difference between LookupDispatchAction and DispatchAction?

Answer :

The difference between LookupDispatchAction and DispatchAction is that the actual method that gets called in LookupDispatchAction is based on a lookup of a key value instead of specifying the method name directly.

Details :

LookupDispatchAction is subclass of DispatchAction.
In case of DispatchAction, you have to declare the method name in the JSP page.
For Example :
http://localhost:8080/emp/empaction.do?step=add // IN CASE OF DispatchAction
here step=add , we are delaring method name in JSP.
or
<html:submit property="step">Add</html:submit> //IN CASE OF DispatchAction
so we can't use localization for button.

To over come both the issues below
a)method name declaration in JSP
b)can't use localization for button
We will go for LookupDispatchAction.

In the LookupDispatchAction :
For example :
If there are three operation like add, delete, update employee.
You can create three different Actions ?
AddEmpAction, DeleteEmpAction and UpdateEmpAction.
This is a valid approach, although not elegant since there might be duplication of code across the Actions since they are related. LookupDispatchAction is the answer to this problem. With LookupDispatchAction, you can combine all three Actions into one.

Example :
Step 1.
three buttons might be
<html:submit property="step">
<bean:message key="button.add"/>
</html:submit>

```
<html:submit property="step">
<bean:message key="button.delete"/>
</html:submit>
<html:submit property="step">
<bean:message key="button.update"/>
</html:submit>
```

//No need method name declaration in JSP . Button name from resource bundle.

Step 2.
In the the Resource Bundle. //Here you can add localization.

```
button.add=Add
button.delete=Delete
button.update=Update
```

Step 3.
Implement a method named getKeyMethodMap() in the subclass of the LookupDispatchAction. The method returns a java.util.Map. The keys used in the Map should be also used as keys in Message Resource Bundle.

```
public class EmpAction extends LookupDispatchAction {

public Map getKeyMethodMap()
{
Map map = new HashMap();
map.put("button.add", "add");
map.put("button.delete", "delete");
map.put("button.update", "update");
}


public ActionForward add(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("add-success");
}

public ActionForward delete(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
```

```
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("delete-success");
}

public ActionForward update(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
//your logic
mapping.findForward("update-success");
}

}
```

in struts-config.xml

```
<action path="/empaction"
input="/empform.jsp"
type="list.EmpAction"
parameter="step"
scope="request"
validate="false">
<forward name="add-success"
path="addEmpSuccess.jsp"
redirect="true"/>
<forward name="delete-success"
path="deleteEmpSuccess.jsp"
redirect="true"/>
<forward name="update-success"
path="updateEmpSuccess.jsp"
redirect="true"/>
</action>
```

for every form submission, LookupDispatchAction does the
reverse lookup on the resource bundle to get the key and then gets the method
whose name is associated with the key from
getKeyMethodmap().

Flow of LookupDispatchAction:

a) Form the button name "Add" get the key "button.add" fro resource bundle.
b) then in the Map getKeyMethodMap() find the value associated with the key "button.add".
c) value from the Map is "add" . then call add() method of the same action class.

more details :
[URL="http://www.techforum360.com/forum/viewthread?thread=135"]http://www.techforum360.com/forum/viewthread?thread=135[/URL]

Question No : 50
What is the use of LookupDispatchAction?
Answer :
LookupDispatchAction is useful if the method name in the Action is not driven by its name in the front end, but by the Locale independent key into the resource bundle. Since the key is always the same, the LookupDispatchAction shields your application from the side effects of I18N.

Question No : 51
What is LookupDispatchAction?
Answer :
The LookupDispatchAction is a subclass of DispatchAction. It does a reverse lookup on the resource bundle to get the key and then gets the method whose name is associated with the key into the Resource Bundle.

Question No : 52
What is the difference between ForwardAction and IncludeAction?
Answer :
The difference is that you need to use the IncludeAction only if the action is going to be included by another action or jsp. Use ForwardAction to forward a request to another resource in your application, such as a Servlet that already does business logic processing or even another JSP page.

Question No : 53
What is IncludeAction?
Answer :
The IncludeAction class is useful when you want to integrate Struts into an application that uses Servlets. Use the IncludeAction class to include another resource in the response to the request being processed.

IncludeAction included resulting resource in the HTTP response .
In the JSP which don't use struts for include we do
<jsp:include page="/test/ServletA"/>

In struts JSP
<jsp:include page="/App1/legacyA.do" />

In the struts-config.xml
<action path="/legacyA"
parameter="/test/ServletA"
type="org.apache.struts.actions.IncludeAction" />

Question No : 54
What is the use of ForwardAction?
Answer :
The ForwardAction class is useful when you?re trying to integrate Struts into an existing application that uses Servlets to perform business logic functions. You can use this class to take advantage of the Struts controller and its functionality, without having to rewrite the existing Servlets. Use ForwardAction to forward a request to another resource in your application, such as a Servlet that already does business logic processing or even another JSP page. By using this predefined action, you don?t have to write your own Action class. You just have to set up the struts-config file properly to use ForwardAction
---
Question No : 55
What is DispatchAction?
Answer :
The DispatchAction class is used to group related actions into one class. Using this class, you can have a method for each logical action compared than a single execute method. The DispatchAction dispatches to one of the logical actions represented by the methods. It picks a method to invoke based on an incoming request parameter. The value of the incoming parameter is the name of the method that the DispatchAction will invoke.

Question No : 56
What are the different kinds of actions in Struts?
Answer :
The different kinds of actions in Struts are:

ForwardAction
IncludeAction
DispatchAction
LookupDispatchAction
SwitchAction

Question No : 57
What is the difference between session scope and request scope when saving formbean ?
Answer :
when the scope is request,the values of formbean would be available for the current request.
when the scope is session,the values of formbean would be available throughout the session.

Question No : 58
Can we have more than one struts-config.xml file for a single Struts application?
Answer :
Yes, we can have more than one struts-config.xml for a single Struts application. They can be configured as follows:

<servlet>
<servlet-name>action</servlet-name> <servlet-class> org.apache.struts.action.ActionServlet
</servlet-class>
<init-param>
<param-name>config</param-name>
<param-value> /WEB-INF/struts-config1.xml, /WEB-INF/struts-config2.xml,
/WEB-INF/struts-config3
</param-value>
</init-param>
<servlet>

Question No : 59
In which method of Action class the business logic is executed ?
Answer :
In the execute() method of Action class the business logic is executed.

public ActionForward execute(ActionMapping mapping, ActionForm form,
HttpServletRequest req, HttpServletResponse res) throws Exception ;

execute() method of Action class:

Perform the processing required to deal with this request
Update the server-side objects (Scope variables) that will be used to create the next page of the

user interface
Return an appropriate ActionForward object

Question No : 60
What is role of Action Class?
Answer :
An Action class in the struts application extends Struts 'org.apache.struts.action.Action" Class.

Action class acts as wrapper around the business logic and provides an inteface to the application's Model layer.

An Action works as an adapter between the contents of an incoming HTTP request and the business logic that corresponds to it.

Then the struts controller (ActionServlet) slects an appropriate Action and Request Processor creates an instance if necessary,

and finally calls execute method of Action class.

To use the Action, we need to Subclass and overwrite the execute() method. and your bussiness login in execute() method.

The return type of the execute method is ActionForward which is used by the Struts Framework to forward the request to the JSP as per the value of the returned ActionForward object.

ActionForward JSP from struts_config.xml file.

Developing our Action Class :

Our Action class (EmpAction.java) is simple class that only forwards the success.jsp.

Our Action class returns the ActionForward called "success", which is defined in the struts-config.xml file (action mapping is show later in this page).

Here is code of our Action Class

public class EmpAction extends Action

```
{
public ActionForward execute(
ActionMapping mapping,
ActionForm form,
HttpServletRequest request,
HttpServletResponse response) throws Exception{
return mapping.findForward("success");
}
}
```

mapping.findForward("success"); forward to JSP mentioned in struts_config.xml.

struts_config.xml configuration is :

```
<action
path="/EmpAction"
type="com.techfaq.EmpAction">
<forward name="success" path="/success.jsp"/>
</action>
```

mapping.findForward("success") method forward to success.jsp (mentioned in struts_config.xml);

Here is the signature of the execute() method Action Class.

public ActionForward execute(ActionMapping mapping,

ActionForm form,

javax.servlet.http.HttpServletRequest request,

javax.servlet.http.HttpServletResponse response)

throws java.lang.Exception

Where

mapping - The ActionMapping used to select this instance

form - The optional ActionForm bean for this request (if any)

request - The HTTP request we are processing

response - The HTTP response we are creating

Throws:

Action class throws java.lang.Exception - if the application business logic throws an exception

In the browser : http://localhost:8080/testApp/EmpAction.do

This will call to execute() method of EmpAction and after that based on mapping.findForward("success") forward to success.jsp.

Question No : 61

How is the Action Mapping specified ?

Answer :
We can specify the action mapping in the configuration file called struts-config.xml. Struts framework creates ActionMapping object from <ActionMapping> configuration element of struts-config.xml file

```
<action-mappings>
<action path="/logon" type="submit.LogonAction"  name="submitForm"
input="/submit.jsp"  scope="request"  validate="true">  <forward name="success"
path="/success.jsp"/>  <forward name="failure" path="/error.jsp"/> </action>
</action-mappings>
```

Question No : 62
What is ActionMapping?
Answer :
ActionMapping object contains all the mapping information in struts_config.xml.

ActionServlet create the instance of ActionMapping and load all the struts_config.xml data to ActionMapping object.

Action mapping contains all the deployment information for a particular Action bean. This class is to determine where the results of the Action will be sent once its processing is complete.

ActionMapping object contains all the mapping information in struts_config.xml.

For Example :

```
<action-mappings>
<action path="/submitForm"
type="com.techfaq.action.EmpAction"
name="EmpForm"
scope="request"
validate="true" // This need for validation framework to validate the form input
input="EmpForm.jsp">
<forward name="success"
path="success.jsp"/>
</action>
```

Question No : 63
Describe validate() and reset() methods ?
Answer :
validate() : Used to validate properties after they have been populated; Called before FormBean is handed to Action. Returns a collection of ActionError as ActionErrors. Following is the method signature for the validate() method.

```
public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {

ActionErrors errors = new ActionErrors();
if ( StringUtils.isNullOrEmpty(username) && StringUtils.isNullOrEmpty(password)){
    errors.add(ActionErrors.GLOBAL_ERROR, new ActionError("error.usernamepassword.required"));
   }
return errors;
}
```

You have to add the error key from resource bundle.

error.usernamepassword.required : key from resource bundle properties file.

In the JSP , you have to add the below tag to display the error message from validate() method.
<html:errors />

reset(): reset() method is called by Struts Framework with each request that uses the defined ActionForm. The purpose of this method is to reset all of the ActionForm's data members prior to the new request values being set.

Example :

```
public void reset(ActionMapping mapping, HttpServletRequest request) {

      this.password = null;
this.username = null;
     }
```

Set null for every request.

Question No : 64

What is role of ActionServlet?
Answer :
The class org.apache.struts.action.ActionServlet is the called the ActionServlet. In the the Jakarta Struts Framework this class plays the role of controller. All the requests to the server goes through the controller. Controller is responsible for handling all the requests.

Struts Flow start with ActionServlet then call to process() method of RequestProcessor.

Step 1. Load ActionServlet using load-on-startup and do the following tasks.

Any struts web application contain the ActionServlet configuration in web.xml file.

On load-on-startup the servlet container Instantiate the ActionServlet .

First Task by ActionServlet : The ActionServlet takes the Struts Config file name as an init-param.

At startup, in the init() method, the ActionServlet reads the Struts Config file and load into memory.

Second Task by ActionServlet : If the user types http://localhost:8080/app/submitForm.do in the browser URL bar, the URL will be intercepted and processed by the ActionServlet since the URL has a pattern *.do, with a suffix of "do". Because servlet-mapping is

<servlet-mapping>

<servlet-name>action</servlet-name>

<url-pattern>*.do</url-pattern>

</servlet-mapping>

Third Task by ActionServlet : Then ActionServlet delegates the request handling to another class called RequestProcessor by invoking its process() method.

Step 2. ActionServlet calls process() method of RequestProcessor

---

Question No : 65
What are the core classes of the Struts Framework?
Answer :
Struts follow Model View Controller Architecture. Components is divided based on MVC.

Controller :

The ActionServlet and the collaborating classes are RequestProcessor, ActionForm, Action, ActionMapping and ActionForward in in Controller category.

ActionServlet :

The central component of the Struts Controller is the ActionServlet. It is a concrete class and extends the javax.servlet.HttpServlet. It performs two important things.
1. On startup, its reads the Struts Configuration file and loads it into memory in the init() method.
2. In the doGet() and doPost() methods, it intercepts HTTP request and handles it appropriately.

View :

The View category contains utility classes ? variety of custom tags (html,bean,logic) and JSP.

Model :

Struts does not offer any components in the Model Category. Your Action class contains business logic may be model.

Question No : 66
What are the components of Struts?
Answer :
Struts and Model View Controller (MVC)
Struts follow Model View Controller Architecture. Components is divided based on MVC.

Controller :

The ActionServlet and the collaborating classes are RequestProcessor, ActionForm, Action, ActionMapping and ActionForward in in Controller category.

ActionServlet :

The central component of the Struts Controller is the ActionServlet. It is a concrete class and extends the javax.servlet.HttpServlet. It performs two important things.
1. On startup, its reads the Struts Configuration file and loads it into memory in the init() method.
2. In the doGet() and doPost() methods, it intercepts HTTP request and handles it appropriately.

View :

The View category contains utility classes ? variety of custom tags (html,bean,logic) and JSP.

Model :

Struts does not offer any components in the Model Category. Your Action class contains business logic may be model.

Question No : 67
How you will enable front-end validation based on the xml in validation.xml?

Answer :
The <html:javascript> tag to allow front-end validation based on the xml in validation.xml. For example the code: <html:javascript formName=\"loginForm\" dynamicJavascript=\"true\" staticJavascript=\"true\" /> generates the client side java script for the form \"loginForm\" as defined in the validation.xml file. The <html:javascript> when added in the jsp file generates the client site validation script.

Question No : 68
How you will display validation fail errors on jsp page?

Answer :
Following tag displays all the errors:
<html:errors/>

Question No : 69
Give the Details of XML files used in Validator Framework?

Answer :
The Validator Framework uses two XML configuration files validator-rules.xml and validation.xml.

The validator-rules.xml defines the standard validation routines, these are reusable and used in validation.xml. to define the form specific validations. The validation.xml defines the validations applied to a form bean.

Details :
http://www.techfaq360.com/tutorial/validation_framework.jsp

Question No : 70
What is Struts Validator Framework?

Answer :
Validation Framework provides the functionality to validate the form data. It can be use to validate the data on the client side as well as on the server side.

Struts has a class called ValidatorForm in org.apache.struts.validator package. This is a subclass of ActionForm and implements the validate() method. The validate() method invokes the Commons Validator, executes the rules using the two xml files (validator-rules.xml and validation.xml) and generates ActionErrors using the Message Resources defined in the struts-config.xml.
validator-rules.xml :
The validator-rules.xml file defines the Validator definitions available for a given application. The validator-rules.xml file acts as a template, defining all of the possible Validators that are available to an application.
validation.xml File :
The validation.xml file is where you couple the individual Validators defined in the validator-rules.xml to components within your application.

Follow the below steps to setup Validation Framework in Struts (server side validation).

Step 1. Add validator plugin into struts-config.xml

<plug-in className="org.apache.struts.validator.ValidatorPlugIn"> <set-property property="pathnames" value="/WEB-INF/validator-rules.xml, /WEB-INF/validation.xml"/>
</plug-in>

Step 3. In the JSP page (EmpForm.jsp)- add <html:errors/>

<%@ taglib uri="/WEB-INF/struts-tiles.tld" prefix="tiles" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>

```
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<html:form>
<html:form action="/submitForm.do" method="post">
<html:errors/> // this is for error message from resource bundle in JSP
<html:text property="firstName" size="20" maxlength="50"/>
<html:text property="lastName" size="20" maxlength="50"/>
<html:submit >Save</html:submit>
</html:form>
```

Step 4. Add Message Resources location in struts-config.xml

```
<!-- Message Resources --> <message-resources parameter="application_resource"/>
```

Step 5. In the the Resource Bundle. application_resource.properties file //Here you can add localization.

```
label.firstName=First Name
label.lastName=Last Name
errors.required={0} is required.
```

Step 7. In the EmpForm

```
package com.techfaq.form;
public class EmpForm extends ValidatorForm {
int empId;
String firstName;
String lastName;
public String getFirstName() {
return firstName;
}
public void setFirstName(String firstName) {
this.firstName = firstName;
}
public String getLastName() {
return lastName;
}
public void setLastName(String lastName) {
this.lastName = lastName;
}
public int getEmpId() {
```

```
return empId;
}
public void setEmpId(int empId) {
this.empId = empId;
}
}
```

Step 6. In the validation.xml - The validation.xml file is where you couple the individual Validators defined in the validator-rules.xml to components within your application

```
<form-validation>
<formset>
<form name="empForm">
<field property="firstName" depends="required">
<arg0 key="label.firstName"/>
</field>
<field property="lastName" depends="required"> <arg0 key="label.lastName"/>
</field>
</form>
</formset>
</form-validation>
```

Step 6. In the validator-rules.xml - The validator-rules.xml file defines the Validator definitions available for a given application.

```
<form-validation>
<global>
<validator
name="required"
classname="org.apache.struts.util.StrutsValidator"
method="validateRequired"
methodparams="java.lang.Object,
org.apache.commons.validator.ValidatorAction,
org.apache.commons.validator.Field,
org.apache.struts.action.ActionErrors,
javax.servlet.http.HttpServletRequest"
msg="errors.required"/>
</global>
</form-validation>
```

Step 5. Add Action mapping and form entry into the stuts-confix.xml and validate="true" is for validation framework to validate the form input.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 1.1//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">
<struts-config>
<form-beans>
<form-bean name="empForm" type="com.techfaq.form.EmpForm"> </form-bean>
</form-beans>
<action-mappings>
<action path="/submitForm"
type="com.techfaq.action.EmpAction"
name="empForm"
scope="request"
validate="true" // This need for validation framework to validate the form input
input="EmpForm.jsp">
<forward name="success"
path="success.jsp"/>
<forward name="failure" path="failure.jsp" />
</action>
</action-mappings>
</struts-config>
```

Now in the browser type http://localhost:8080/testApp/EmpForm.jsp

Don't Enter firstName and lastName in the text box and submit the "Save" BUTTON. the RequestProcessor checks for the validateattribute in the ActionMapping.
If the validate is set to true, the RequestProcessor invokes the validate() method of the ValidatorForm instance.
If Validate fail the RequestProcessor looks for the input attribute and return to JSP page mentioned in input tag.
If Validate pass goto Action Class execute() method..
If Validate fail , In the browser (EmpForm.jsp) you can see. In red color.

First Name is required.
Last Name is required.

In the empForm firstName and lastName are the required filed. So in the above configuration you can see we add for both firstName and lastName. You can see depends="required" - "required" property is defind in validator-rules.xml. In the resource bundle : application_resource.propertis file

label.firstName=First Name
label.lastName=Last Name
#Error messages used by the Validator
errors.required={0} is required.
{0} will be filled by (First Name or Last Name) because validation.xml above configuration you have defind
<arg0 key="label.lastName"/>. and <arg0 key="label.lastName"/>.

Question No : 71
What is ActionForm?

Answer :
An ActionForm is a JavaBean that extends org.apache.struts.action.ActionForm. ActionForm maintains the session state for web application and the ActionForm object is automatically populated (by RequestProcessor)on the server side with data entered from a form on the client side.

If your JSP contain two fields
username and password
<html:text property = "username" maxlength = "60" >
<html:password property = "password" maxlength = "60" />

Then the ActionForm is

```
public class LogonForm extends ActionForm{

  /**
     * Password entered by user
     */
    private String password = null;
    /**
     * Username entered by user
     */
    private String username = null;
/**
   * @return Returns the password.
   */
  public String getPassword() {
```

```
   return password;
  }
  /**
   * @param password The password to set.
   */
  public void setPassword(String password) {
   this.password = password;
  }
  /**
   * @return Returns the username.
   */
  public String getUsername() {
   return username;
  }
  /**
   * @param username The username to set.
   */
  public void setUsername(String username) {
   this.username = username;
  }
}
```

When you enter data to your JSP and submit the request processor set the data to LogonForm. You don't need to do request.getParameter("username") etc,

---

Question No : 72
What is Action Class?

Answer :
An Action class in the struts application extends Struts 'org.apache.struts.action.Action" Class.

Action class acts as wrapper around the business logic and provides an inteface to the application's Model layer.

An Action works as an adapter between the contents of an incoming HTTP request and the business logic that corresponds to it.

Then the struts controller (ActionServlet) slects an appropriate Action and Request Processor creates an instance if necessary,

and finally calls execute method of Action class.

To use the Action, we need to Subclass and overwrite the execute() method. and your bussiness login in execute() method.

The return type of the execute method is ActionForward which is used by the Struts Framework to forward the request to the JSP as per the value of the returned ActionForward object.

ActionForward JSP from struts_config.xml file.

Developing our Action Class :

Our Action class (EmpAction.java) is simple class that only forwards the success.jsp.

Our Action class returns the ActionForward called "success", which is defined in the struts-config.xml file (action mapping is show later in this page).

Here is code of our Action Class

```
public class EmpAction extends Action

{

public ActionForward execute(

ActionMapping mapping,

ActionForm form,

HttpServletRequest request,

HttpServletResponse response) throws Exception{

return mapping.findForward("success");

}

}
```

mapping.findForward("success"); forward to JSP mentioned in struts_config.xml.

struts_config.xml configuration is :

```
<action
path="/EmpAction"
type="com.techfaq.EmpAction">
<forward name="success" path="/success.jsp"/>
</action>
```

mapping.findForward("success") method forward to success.jsp (mentioned in struts_config.xml);

Here is the signature of the execute() method Action Class.

```
public ActionForward execute(ActionMapping mapping,
ActionForm form,
javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception
```

Where

mapping - The ActionMapping used to select this instance

form - The optional ActionForm bean for this request (if any)

request - The HTTP request we are processing

response - The HTTP response we are creating

Throws:

Action class throws java.lang.Exception - if the application business logic throws an exception

In the browser : http://localhost:8080/testApp/EmpAction.do

This will call to execute() method of EmpAction and after that based on mapping.findForward("success") forward to success.jsp.

Question No : 73
How you will make available any Message Resources Definitions file to the Struts Framework Environment?

Answer :
Message Resources Definitions file are simple .properties files and these files contains the messages that can be used in the struts project. Message Resources Definitions files can be added to the struts-config.xml file through <message-resources /> tag.
Example:

<message-resources parameter="application_resource"/>

application_resource is properties file.

application_resource.properties looks like
error.vendor.location = Error at Vendor Location.
error.Recordingurl.needed = URL is Mandatory for Other Vendor.

Question No : 74
What is ActionServlet?

Answer :
The class org.apache.struts.action.ActionServlet is the called the ActionServlet. In the the Jakarta Struts Framework this class plays the role of controller. All the requests to the server goes through the controller. Controller is responsible for handling all the requests.

Struts Flow start with ActionServlet then call to process() method of RequestProcessor.

Step 1. Load ActionServlet using load-on-startup and do the following tasks.

Any struts web application contain the ActionServlet configuration in web.xml file.
On load-on-startup the servlet container Instantiate the ActionServlet .
First Task by ActionServlet : The ActionServlet takes the Struts Config file name as an init-param.
At startup, in the init() method, the ActionServlet reads the Struts Config file and load into memory.

Second Task by ActionServlet : If the user types http://localhost:8080/app/submitForm.do in the browser URL bar, the URL will be intercepted and processed by the ActionServlet since the URL has a pattern *.do, with a suffix of "do". Because servlet-mapping is
<servlet-mapping>
<servlet-name>action</servlet-name>
<url-pattern>*.do</url-pattern>
</servlet-mapping>
Third Task by ActionServlet : Then ActionServlet delegates the request handling to another class called RequestProcessor by invoking its process() method.

Step 2. ActionServlet calls process() method of RequestProcessor
---
Question No : 75
Q.How you will display validation fail errors on jsp page?

Answer :
Add <html:errors/> to your jsp page.

validate() : Used to validate properties after they have been populated; Called before FormBean is handed to Action. Returns a collection of ActionError as ActionErrors. Following is the method signature for the validate() method.

public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {

ActionErrors errors = new ActionErrors();
if (StringUtils.isNullOrEmpty(username) && StringUtils.isNullOrEmpty(password)){
errors.add(ActionErrors.GLOBAL_ERROR, new ActionError("error.usernamepassword.required"));
}

return errors;
}

You have to add the error key from resource bundle.

error.usernamepassword.required : key from resource bundle properties file.

In the JSP , you have to add the below tag to display the error message from validate() method.
<html:errors />

Question No : 76
Q.How you will enable front-end client validation based on the xml in validation.xml?

Answer :
The <html:javascript> tag to allow front-end validation based on the xml in validation.xml.

Validation Framework provides the functionality to validate the form data. It can be use to validate the data on the client side. Errors will be displayed like java script.

Struts has a class called ValidatorForm in org.apache.struts.validator package. This is a subclass of ActionForm and implements the validate() method. The validate() method invokes the Commons Validator, executes the rules using the two xml files (validator-rules.xml and validation.xml) and generates ActionErrors using the Message Resources defined in the struts-config.xml.
validator-rules.xml :
The validator-rules.xml file defines the Validator definitions available for a given application. The validator-rules.xml file acts as a template, defining all of the possible Validators that are available to an application.
validation.xml File :
The validation.xml file is where you couple the individual Validators defined in the validator-rules.xml to components within your application.

Follow the below steps to setup Validation Framework in Struts (server side validation).

Step 1. Add validator plugin into struts-config.xml

<plug-in className="org.apache.struts.validator.ValidatorPlugIn"> <set-property property="pathnames" value="/WEB-INF/validator-rules.xml, /WEB-INF/validation.xml"/>
</plug-in>

Step 3. In the JSP page (EmpForm.jsp)- Add onsubmit="return validateempForm(this);" and <html:javascript formName="empForm"/> for Client Side Java Script Validation.

```
<%@ taglib uri="/WEB-INF/struts-tiles.tld" prefix="tiles" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<html:form>
<html:form action="/submitForm.do" method="post" onsubmit="return validateempForm(this);">
<html:errors/> // this is for error message from resource bundle in JSP
<html:text property="firstName" size="20" maxlength="50"/>
<html:text property="lastName" size="20" maxlength="50"/>
<html:submit >Save</html:submit>
<!-- Begin Validator Javascript Function-->
<html:javascript formName="empForm"/>
<!-- End of Validator Javascript Function-->
</html:form>
```

Step 4. Add Message Resources location in struts-config.xml

```
<!-- Message Resources --> <message-resources parameter="application_resource"/>
```

Step 5. In the the Resource Bundle. application_resource.properties file //Here you can add localization.

```
label.firstName=First Name
label.lastName=Last Name
errors.required={0} is required.
```

Step 7. In the EmpForm

```
package com.techfaq.form;
public class EmpForm extends ValidatorForm {
int empId;
String firstName;
String lastName;
public String getFirstName() {
return firstName;
}
```

```
public void setFirstName(String firstName) {
this.firstName = firstName;
}
public String getLastName() {
return lastName;
}
public void setLastName(String lastName) {
this.lastName = lastName;
}
public int getEmpId() {
return empId;
}
public void setEmpId(int empId) {
this.empId = empId;
}
}
```

Step 6. In the validation.xml - The validation.xml file is where you couple the individual Validators defined in the validator-rules.xml to components within your application

```
<form-validation>
<formset>
<form name="empForm">
<field property="firstName" depends="required">
<arg0 key="label.firstName"/>
</field>
<field property="lastName" depends="required"> <arg0 key="label.lastName"/>
</field>
</form>
</formset>
</form-validation>
```

Step 6. In the validator-rules.xml - The validator-rules.xml file defines the Validator definitions available for a given application.

```
<form-validation>
<global>
<validator
name="required"
classname="org.apache.struts.util.StrutsValidator"
method="validateRequired"
```

```
methodparams="java.lang.Object,
org.apache.commons.validator.ValidatorAction,
org.apache.commons.validator.Field,
org.apache.struts.action.ActionErrors,
javax.servlet.http.HttpServletRequest"
msg="errors.required"/>
</global>
</form-validation>
```

Step 5. Add Action mapping and form entry into the stuts-confix.xml and validate="true" is for validation framework to validate the form input.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 1.1//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">
<struts-config>
<form-beans>
<form-bean name="EmpForm" type="com.techfaq.form.EmpForm"> </form-bean>
</form-beans>
<action-mappings>
<action path="/submitForm"
type="com.techfaq.action.EmpAction"
name="EmpForm"
scope="request"
validate="true" // This need for validation framework to validate the form input
input="EmpForm.jsp">
<forward name="success"
path="success.jsp"/>
<forward name="failure" path="failure.jsp" />
</action>
</action-mappings>
</struts-config>
```

Now in the browser type http://localhost:8080/testApp/EmpForm.jsp

Don't Enter firstName and lastName in the text box and submit the "Save" BUTTON. the RequestProcessor checks for the validateattribute in the ActionMapping.
If the validate is set to true, the RequestProcessor invokes the validate() method of the ValidatorForm instance.

If Validate fail the RequestProcessor looks for the input attribute and return to JSP page mentioned in input tag.
If Validate pass goto Action Class execute() method..
If Validate fail , In the browser (EmpForm.jsp) you can see. Java Script pop up Message:

First Name is required.
Last Name is required.

In the empForm firstName and lastName are the required filed. So in the above configuration you can see we add for both firstName and lastName. You can see depends="required" - "required" property is defind in validator-rules.xml. In the resource bundle : application_resource.propertis file

label.firstName=First Name
label.lastName=Last Name
#Error messages used by the Validator
errors.required={0} is required.
{0} will be filled by (First Name or Last Name) because validation.xml above configuration you have defind
<arg0 key="label.lastName"/>. and <arg0 key="label.lastName"/>.

****

****
Spring Interview Questions
****
Question No : 1
How do you integrate Struts Spring and Hibernate ?
Answer :
Code is here with explanation

[URL=http://www.techfaq360.com/tutorial/spring/struts_spring_hibernate.jsp]http://www.techfaq360.com/tutorial/spring/struts_spring_hibernate.jsp[/URL]

Question No : 2
How do you Integrate Spring with Hibernate ?
Answer :

Steps to Integrate Spring with Hibernate
[URL=http://www.techfaq360.com/tutorial/spring/springhibernate.jsp]http://www.techfaq360.com/tutorial/spring/springhibernate.jsp[/URL]

Question No : 3
How Spring relate to MVC framework ?
Answer :
In Spring's Web MVC framework : a DispatcherServlet that dispatches requests to handlers.The default handler is a very simple Controller interface, just offering a ModelAndView handleRequest(request,response) method.
Details with code
[URL=http://www.techfaq360.com/tutorial/spring/webmvc.jsp]http://www.techfaq360.com/tutorial/spring/webmvc.jsp[/URL]

Question No : 4
How Validation Framework work in Spring with code Example ?
Answer :
get details at
[URL=http://www.techfaq360.com/tutorial/spring/springvalidation.jsp]http://www.techfaq360.com/tutorial/spring/springvalidation.jsp[/URL]

Question No : 5
How to setup MessageSources in Spring ?
Answer :
Spring currently provides two MessageSource implementations. These are the ResourceBundleMessageSource and the StaticMessageSource.
Details with code example
[URL=http://www.techfaq360.com/tutorial/spring/messageresource.jsp]http://www.techfaq360.com/tutorial/spring/messageresource.jsp[/URL]

Question No : 6
What is ApplicationContext in details ?
Answer :
The basis for the context package is the ApplicationContext interface, located in the org.springframework.context package. Deriving from the BeanFactory interface, it provides all the functionality of BeanFactory.

[URL=http://www.techfaq360.com/tutorial/spring/applicationcontext.jsp]http://www.techfaq360.com/tutorial/spring/applicationcontext.jsp[/URL]

Question No : 7
Explain Bean Lifecycle in Spring ?

Answer :
The Spring Framework provides InitializingBean and DisposableBean marker interface. Implementing these interfaces will result in the container calling afterPropertiesSet() for the former and destroy() for the latter to allow the bean to perform certain actions upon initialization and destruction.

Details are here
[URL=http://www.techfaq360.com/tutorial/spring/beanlifecycle.jsp]http://www.techfaq360.com/tutorial/spring/beanlifecycle.jsp[/URL]

Question No : 8
What are the Bean Scopes in Spring ?
Answer :
[URL=http://www.techfaq360.com/tutorial/spring/beanscopes.jsp]http://www.techfaq360.com/tutorial/spring/beanscopes.jsp[/URL]

Question No : 9
How spring Setter Injection work with List, Set, Map and Properties ?

Answer :
Example is here
[URL=http://www.techfaq360.com/tutorial/spring/collectioninjection.jsp]http://www.techfaq360.com/tutorial/spring/collectioninjection.jsp[/URL]

Question No : 10
Explain Dependency Injection with code Example ?
Answer :
The basic principle behind Dependency Injection (DI) is that objects define their dependencies . Then, it is the job of the container to actually inject those dependencies when it creates the bean. The basic concept of the Inversion of Control pattern (also known as Dependency Injection (DI)) is that you do not create your objects but describe how they should be created. You don't directly connect your components and services together in code but describe which services are needed by which components in a configuration file. A container (in the case of the Spring framework, the IOC container) is then responsible for hooking it all up. In a typical IOC scenario, the container creates all the objects, wires them together by setting the necessary properties, and determines when methods will be invoked.

The two major flavors of Dependency Injection are Setter Injection (injection via JavaBean setters) and Constructor Injection (injection via constructor arguments).

Get the code with example

[URL=http://www.techfaq360.com/tutorial/spring/dependencyinjection.jsp]http://www.techfaq360.com/tutorial/spring/dependencyinjection.jsp[/URL]

Question No : 11

What is IOC-Inversion of Control ?

Answer :

The org.springframework.beans and org.springframework.context packages provide the basis for the Spring Framework's IoC container.

Details you can get at
[URL=http://www.techfaq360.com/tutorial/spring/ioc.jsp]http://www.techfaq360.com/tutorial/spring/ioc.jsp[/URL]

Question No : 12

What do you mean by Advice? and what are the Advices?

Answer :

Action taken by an aspect at a particular join point. Different types of advice include "around," "before" and "after" advice. Many AOP frameworks, including Spring, model an advice as an interceptor, maintaining a chain of interceptors "around" the join point.

Types of advice:

Before advice: Advice that executes before a join point, but which does not have the ability to prevent execution flow proceeding to the join point (unless it throws an exception).

After returning advice: Advice to be executed after a join point completes normally: for example, if a method returns without throwing an exception.

After throwing advice: Advice to be executed if a method exits by throwing an exception.

After (finally) advice: Advice to be executed regardless of the means by which a join point exits (normal or exceptional return).

Around advice: Advice that surrounds a join point such as a method invocation. This is the most powerful kind of advice. Around advice can perform custom behavior before and after the method

invocation. It is also responsible for choosing whether to proceed to the join point or to shortcut the advised method execution by returning its own return value or throwing an exception

---

Question No : 13

What do you mean by JointPoint?

Answer :

A point during the execution of a program, such as the execution of a method or the handling of an exception. In Spring AOP, a join point always represents a method execution.

Question No : 14

How to integrate Struts Spring Hibernate ?

Answer :

Step 1.

In the struts-config.xml add plugin

```
<plug-in className="org.springframework.web.struts.ContextLoaderPlugIn">
        <set-property property="contextConfigLocation"
            value="/WEB-INF/applicationContext.xml"/>
    </plug-in>
```

Step 2.

In the applicationContext.xml file

Configure datasourse

```
    <bean id="dataSource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
     <property name="driverClassName"><value>oracle.jdbc.driver.OracleDriver</value>
      </property>
        <property name="url"><value>jdbc:oracle:thin:@10.10.01.24:1541:ebizd</value>
        </property>
        <property name="username"><value>sa</value></property>
   <property name="password"><value></value></property>
    </bean>
```

Step 3.

Configure SessionFactory

```
 <!-- Hibernate SessionFactory -->
    <bean id="sessionFactory"
class="org.springframework.orm.hibernate.LocalSessionFactoryBean">
        <property name="dataSource"><ref local="dataSource"/></property>
```

```
        <property name="mappingResources">
            <list>
                <value>com/test/dbxml/User.hbm.xml</value>
            </list>
        </property>
        <property name="hibernateProperties">
        <props>
            <prop key="hibernate.dialect">net.sf.hibernate.dialect.OracleDialect </prop>
         </props>
        </property>
    </bean>
```

Step 4.
Configure User.hbm.xml

```
<hibernate-mapping>
    <class name="org.test.model.User" table="app_user">

        <id name="id" column="id" >
            <generator class="increment"/>
        </id>
        <property name="firstName" column="first_name" not-null="true"/>
        <property name="lastName" column="last_name" not-null="true"/>

    </class>
</hibernate-mapping>
```

Step 5.

In the applicationContext.xml ? configure for DAO

```
 <bean id="userDAO" class="org.test.dao.hibernate.UserDAOHibernate">
        <property name="sessionFactory"><ref local="sessionFactory"/></property>
    </bean>
```

Step 6.

DAO Class

```
public class UserDAOHibernate extends HibernateDaoSupport implements UserDAO {
    private static Log log = LogFactory.getLog(UserDAOHibernate.class);
```

```
    public List getUsers() {
        return getHibernateTemplate().find("from User");
    }

    public User getUser(Long id) {
        return (User) getHibernateTemplate().get(User.class, id);
    }

    public void saveUser(User user) {
        getHibernateTemplate().saveOrUpdate(user);

        if (log.isDebugEnabled()) {
            log.debug("userId set to: " + user.getId());
        }
    }

    public void removeUser(Long id) {
        Object user = getHibernateTemplate().load(User.class, id);
        getHibernateTemplate().delete(user);
    }
}
```

Question No : 15
What are the types of Advice?

Answer :
Types of advice:

Before advice: Advice that executes before a join point, but which does not have the ability to prevent execution flow proceeding to the join point (unless it throws an exception).

After returning advice: Advice to be executed after a join point completes normally: for example, if a method returns without throwing an exception.

After throwing advice: Advice to be executed if a method exits by throwing an exception.

After (finally) advice: Advice to be executed regardless of the means by which a join point exits

(normal or exceptional return).

Around advice: Advice that surrounds a join point such as a method invocation. This is the most powerful kind of advice. Around advice can perform custom behavior before and after the method invocation. It is also responsible for choosing whether to proceed to the join point or to shortcut the advised method execution by returning its own return value or throwing an exception

---

Question No : 16

How to integrate Spring with Hibernate ?

Answer :

There are two steps

step 1.

In the bean.xml which is realted to bean factory spring.

```
<beans>
   <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
      <property name="driverClassName">
         <value>org.hsqldb.jdbcDriver</value>
      </property>
      <property name="url">
         <value>jdbc:hsqldb:data/tutorial</value>
      </property>
      <property name="username">
         <value>sa</value>
      </property>
      <property name="password">
         <value></value>
      </property>
   </bean>
   <bean id="factory" class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
      <property name="mappingResources">
         <list>
            <value>Event.hbm.xml</value>
         </list>
      </property>
      <property name="hibernateProperties">
         <props>
            <prop key="hibernate.dialect">org.hibernate.dialect.HSQLDialect</prop>
            <prop key="hibernate.show_sql">false</prop>
         </props>
      </property>
```

```
        <property name="dataSource">
            <ref bean="dataSource"/>
        </property>
    </bean>
    <bean id="eventDao" class="EventSpringDao">
        <property name="sessionFactory">
            <ref bean="factory" />
        </property>
    </bean>
</beans>
```

EventSpringDao class has setter and getter method of sessionFactory.

Step 2.

In the EventSpringDao class you can do getSessionFactory().openSession() to access data base.

or
your class you can extend HibernateDaoSupport.
Exmaple:

```
public EventSpringDao extends HibernateDaoSupport{
 public void saveOrUpdate(Event event){
        getHibernateTemplate().saveOrUpdate(event);
    }

}
```

Question No : 17
What is JdbcTemplate in Spring ?
Answer :
The JdbcTemplate class is the central class in the JDBC core package. It simplifies the use of JDBC since it handles the creation and release of resources. This helps to avoid common errors such as forgetting to always close the connection. It executes the core JDBC workflow like statement creation and execution, leaving application code to provide SQL and extract results. This class executes SQL queries, update statements or stored procedure calls, imitating iteration over ResultSets and extraction of returned parameter values. It also catches JDBC exceptions and translates them to the generic, more informative, exception hierarchy defined in the org.springframework.dao package.

In the bean.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">

   <bean id="eventDao" class="com.example.JdbcEventDao">
      <property name="dataSource" ref="dataSource"/>
   </bean>

   <!-- the DataSource (parameterized for configuration via a PropertyPlaceHolderConfigurer) -->
   <bean id="dataSource" destroy-method="close"
class="org.apache.commons.dbcp.BasicDataSource">
      <property name="driverClassName" value="${jdbc.driverClassName}"/>
      <property name="url" value="${jdbc.url}"/>
      <property name="username" value="${jdbc.username}"/>
      <property name="password" value="${jdbc.password}"/>
   </bean>

</beans>
```

And In Java class

```
public class JdbcEventDao implements EventDao {

   private JdbcTemplate jdbcTemplate;

   public void setDataSource(DataSource dataSource) {
      this.jdbcTemplate = new JdbcTemplate(dataSource);
   }

   /// DO METHODS
}
```

Question No : 18

What is AOP - Aspect-oriented programming ?

Answer :

the system service like logging, transaction management, security etc., must be included in the program in general case.

AOP makes it possible to modularize and separate these services and then apply them

declaratively to the components and we can focus on our own specific concerns.
Transaction parameters are like EJB
PROPAGATION_REQUIRED
PROPAGATION_REQUIRED_NEW etc

Question No : 19
What is RowCallbackHandler ?

Answer :
The RowCallbackHandler interface extracts values from each row of a ResultSet.

Has one method ? processRow(ResultSet)
Called for each row in ResultSet.

Question No : 20
What is PreparedStatementCreator ?

Answer :
PreparedStatementCreator:

Is one of the most common used interfaces for writing data to database.
Has one method ? createPreparedStatement(Connection)
Responsible for creating a PreparedStatement.
Does not need to handle SQLExceptions.

Question No : 21
What are Bean scopes in Spring Framework ?
Answer :
If you are using ApplicationContext then five scopes (singleton,prototype,request,session,globalsession) and if you are using BeanFactory then two(singleton,prototype);

singleton :
 Scopes a single bean definition to a single object instance per Spring IoC container. Only one singleton object will be created. Default scope.

prototype:
 Scopes a single bean definition to any number of object instances. Every call create new object like Emp emp = new Emp();

request:
Scopes a single bean definition to the lifecycle of a single HTTP request; that is each and every HTTP request will have its own instance of a bean created off the back of a single bean definition. Only valid in the context of a web-aware Spring ApplicationContext.

session:
Scopes a single bean definition to the lifecycle of a HTTP Session. Only valid in the context of a web-aware Spring ApplicationContext.

global session :
Scopes a single bean definition to the lifecycle of a global HTTP Session. Typically only valid when used in a portlet context. Only valid in the context of a web-aware Spring ApplicationContext.

Question No : 22
Q.Lifecycle interfaces in spring ?
Answer :
1) InitializingBean
2) DisposableBean

1) InitializingBean
<bean id="exampleInitBean" class="examples.ExampleBean" init-method="init"/>
public class ExampleBean {
public void init() { // do some initialization work }
}
OR
<bean id="exampleInitBean" class="examples.AnotherExampleBean"/>
public class AnotherExampleBean implements InitializingBean {
public void afterPropertiesSet() { // do some initialization work }
}

2) DisposableBean
<bean id="exampleInitBean" class="examples.ExampleBean" destroy-method="cleanup"/>
public class ExampleBean {

```
 public void cleanup() {
 // do some destruction work (like releasing pooled connections)
 } }
```

OR

```
<bean id="exampleInitBean" class="examples.AnotherExampleBean"/>
public class AnotherExampleBean implements DisposableBean {
public void destroy() { // do some destruction work (like releasing pooled connections) }
 }
```

Question No : 23

Q.What are the common implementations of the Application Context ?

Answer :

* ClassPathXmlApplicationContext : It Loads context definition from an XML file located in the classpath

```
ApplicationContext context = new ClassPathXmlApplicationContext("bean_test.xml");
```

*FileSystemXmlApplicationContext : It loads context definition from an XML file in the filesystem.

```
ApplicationContext context = new FileSystemXmlApplicationContext("bean_test.xml");
```

XmlWebApplicationContext : It loads context definition from an XML file contained within a web application.

Question No : 24

Q.What is the difference between Bean Factory and Application Context ?

Answer :

*The ApplicationContext builds on top of the BeanFactory (it's a subclass) and adds other functionality such as easier integration with Springs AOP features, message resource handling (for use in internationalization), event propagation, declarative mechanisms to create the ApplicationContext and optional parent contexts, and application-layer specific contexts such as the WebApplicationContext, among other enhancements.

*Application Context is suitabe for J2EE Applications.

*The main usage scenario when you might prefer to use the BeanFactory is when memory usage is the greatest concern (such as in an applet where every last kilobyte counts), and you don't need

all the features of the ApplicationContext.

Question No : 25
Q.What is Application Context in Spring?

Answer :
The ApplicationContext builds on top of the BeanFactory (it's a subclass) and adds other functionality such as easier integration with Springs AOP features, message resource handling (for use in internationalization), event propagation, declarative mechanisms to create the ApplicationContext and optional parent contexts, and application-layer specific contexts such as the WebApplicationContext, among other enhancements.

the BeanFactory provides the configuration framework and basic functionality, while the ApplicationContext adds enhanced capabilities to it, some of them perhaps more J2EE and enterprise-centric. In general, an ApplicationContext is a complete superset of a BeanFactory, and any description of BeanFactory capabilities and behavior should be considered to apply to ApplicationContexts as well.

in a J2EE-environment, the best option is to use the ApplicationContext, since it offers all the features of the BeanFactory and adds on to it in terms of features, while also allowing a more declarative approach to use of some functionality, which is generally desirable.

Question No : 26
Q.What is BeanFactory and how can configure BeanFactory ?
Answer :
The BeanFactory is the actual container which instantiates, configures, and manages a number of beans. These beans typically collaborate with one another, and thus have dependencies between themselves. These dependencies are reflected in the configuration data used by the BeanFactory.

Three ways you can configure BeanFactory.
Resource res = new FileSystemResource("beans.xml");
XmlBeanFactory factory = new XmlBeanFactory(res);
or

ClassPathResource res = new ClassPathResource("beans.xml");
XmlBeanFactory factory = new XmlBeanFactory(res);
or

ClassPathXmlApplicationContext appContext = new ClassPathXmlApplicationContext(

```
        new String[] {"applicationContext.xml", "applicationContext-part2.xml"});
// of course, an ApplicationContext is just a BeanFactory
BeanFactory factory = (BeanFactory) appContext;
```

Question No : 27

What are the types of Dependency Injection Spring supports with Example?

Answer :

Setter Injection and Constructor Injection .

The two major flavors of Dependency Injection are Setter Injection (injection via JavaBean setters); and Constructor Injection (injection via constructor arguments). Spring provides sophisticated support for both, and even allows you to mix the two when configuring the one object.

Setter Injection Example :

XML ----

```
<bean id="createCreditCard"
class="springexample.creditcardaccount.CreateCreditCardAccount">
<property name="smsInterface">
   <ref bean="sms" />
  </property>
  <property name="daoInterface">
   <ref bean="dao" />
  </property>
</bean>
```

Java Class : CreateCreditCardAccount : Setter and Getter

```
public SMSInterface getSmsInterface() {
  return smsInterface;
 }
 /**
  * @param smsInterface The smsInterface to set.
  */
 public void setSmsInterface(SMSInterface smsInterface) {
  this.smsInterface = smsInterface;
 }
 /**
  * @return Returns the daoInterface.
  */
 public DAOInterface getDaoInterface() {
  return daoInterface;
```

```
 }
 /**
  * @param daoInterface The daoInterface to set.
  */
 public void setDaoInterface(DAOInterface daoInterface) {
  this.daoInterface = daoInterface;
 }
```

Constructor Injection Example:

```
 <bean id="orderService"
        class="spring.OrderService">
        <constructor-arg>
            <ref bean="orderDAO">
        </constructor-arg>
    </bean>
```

Java Class : OrderService

```
public class OrderService{
private OrderDAO orderDAO;
  public OrderService(OrderDAO orderDAO){
   this.orderDAO=orderDAO;
}
}
```

Question No : 28

What is Spring ? and What are the Benefits/Advantages of Using Spring Framework ?

Answer :

Spring is an open source framework created to address the complexity of enterprise application development.

*Lightweight container

*No App Server Dependent ? like EJB JNDI Calls

*Objects are created Lazily , Singleton - configuration

*Components can added Declaratively

Initialization of properties is easy ? no need to read from properties file

*Declarative transaction, security and logging service - AOP

application code is much easier to unit test

*With a Dependency Injection approach, dependencies are explicit, and evident in constructor or JavaBean properties
*Spring's configuration management services can be used in any architectural layer, in whatever runtime environment.
*Spring can effectively organize your middle tier objects
*not requires special deployment steps

Question No : 29
What are the benefits of IOC - Inversion of control (Dependency Injection)?

Answer :
*Loose coupling: Components can added Declaratively so we can add and remove the components with out code change.
For Example :

```
<bean id="createCreditCard"
class="springexample.creditcardaccount.CreateCreditCardAccount">
  <property name="emailInterface">
   <ref bean="email" />
  </property>
  <property name="smsInterface">
   <ref bean="sms" />
  </property>
  <property name="daoInterface">
   <ref bean="dao" />
  </property>

 </bean>
```

There are three components associated with createCreditCard.
Tommorow you don't want smsInterface so just remove the xml configuration.

```
  <property name="smsInterface">
   <ref bean="sms" />
  </property>
```

*Not Required any singletons : Don't need to code for singleton class. Every class is by default singleton. you can make not singleton
by making singleton="false"

```
<bean id="sms" class="springexample.sms.SMS" singleton="false">
</bean>
```

*Objects are created Lazily .

*Initialization of properties is easy ? no need to read from properties file.
For Example:

```
<bean id="email" class="springexample.email.Email" >
  <property name="smtpHost">
    <value>smtp.sa.com</value>
  </property>
</bean>
```

IOC Contained set the setter method of Email class. setSmtpHost(). You don't nned to read from properties file with extra coding.

*No App Server Dependent ? like EJB JNDI Calls

Question No : 30
What are the different types of IOC (dependency injection) ?

Answer :
Spring container support
Setter Injection and Constructor Injection .

The two major flavors of Dependency Injection are Setter Injection (injection via JavaBean setters); and Constructor Injection (injection via constructor arguments). Spring provides sophisticated support for both, and even allows you to mix the two when configuring the one object.

Interface Injection (e.g. Avalon): Injection is done through an interface. Spring does not support this.

Question No : 31
Q;What is IOC - Inversion of Control ?
Answer :
The basic concept of the Inversion of Control pattern (also known as dependency injection) is that you do not create your objects but describe how they should be created.
For Example :
In the XML file you declare

```
<bean id="creditRating" class="springexample.creditrating.CreditRating">
   </bean>
```

You don't need to create object for CreditRating.
No nned to do new CreditRating(). Spring container will create the object for you.

You don't directly connect your components and services together in code but describe which services are needed by which components in a configuration file. A container (in the case of the Spring framework, the IOC container) is then responsible for hooking it all up.
In a typical IOC scenario, the container creates all the objects, wires them together by setting the necessary properties, and determines when methods will be invoked.

For Example :
In the XML File

```
<bean id="createCreditCard"
class="springexample.creditcardaccount.CreateCreditCardAccount">
  <property name="daoInterface">
   <ref bean="dao" />
  </property>

 </bean>
```

Then Spring container will create the dependency relation. CreateCreditCardAccount class should have setter and getter for daoInterface.

You don't need to create daoInterface object and set the setter method. Spring container will do for you.

The two major flavors of Dependency Injection are Setter Injection (injection via JavaBean setters); and Constructor Injection (injection via constructor arguments). Spring provides sophisticated support for both, and even allows you to mix the two when configuring the one object

---
Question No : 32
What is the Benefits/Advantages of Using Spring Framework ?
Answer :
*Lightweight container
*No App Server Dependent ? like EJB JNDI Calls
Objects are created Lazily , Singleton - configuration
*Components can added Declaratively
*Initialization of properties is easy ? no need to read from properties file
*Declarative transaction, security and logging service - AOP
*application code is much easier to unit test
With a Dependency Injection approach, dependencies are explicit, and evident in constructor or

JavaBean properties
*Spring's configuration management services can be used in any architectural layer, in whatever runtime environment.
*Spring can effectively organize your middle tier objects
*requires no special deployment steps

Question No : 33
Q.what is dependency injection ??
Answer :
The basic concept of the Inversion of Control pattern (also known as dependency injection) is that you do not create your objects but describe how they should be created. You don't directly connect your components and services together in code but describe which services are needed by which components in a configuration file.

****

****
JDBC Interview Questions
****
Question No : 1
Image upload from JSP to DataBase ?
Answer :
Image upload from JSP to Data Base step by step :

step 1 :
In the JSP
<form name="regform2" method="post" enctype="multipart/form-data">
<input type="file" name="ImageFile" id="ImageFile" onChange="uploadImage()"/>
</form>

java script :
function uploadImage(){
document.regform.action ="<%=request.getContextPath()%>/dfdmin?cmd=uploadimage";
document.regform.submit();
}

Step 2.
In the servlet - add the below code.
String rtempfile = File.createTempFile("temp","1").getParent();
MultipartRequest multi = new MultipartRequest(request, rtempfile, 500000 * 1024);

```
File rnewfile=null;
rnewfile = new
File(CommonArt.IMAGE_PATH+"jsp"+File.separator+"images"+File.separator+"uploadImage"+Fil
e.separator);
if(rnewfile.exists()){
}else{
rnewfile.mkdirs();
}

File f = multi.getFile("ImageFile");
System.out.println(f.getName());
FileInputStream fin =new FileInputStream(f);
RandomAccessFile r = new RandomAccessFile(rnewfile+File.separator+f.getName(),"rw");
filename = f.getName();
// FileOutputStream fos =new FileOutputStream(rnewfile);
byte sizefile[] = new byte[5000000];
fin.read(sizefile);
// fos.write(sizefile);
r.write(sizefile);
//fos.close();
r.close();
fin.close();
```

Step 3.
Insert into Database

```
InputStream is = new FileInputStream(f);
String sql = " INSERT INTO image_upload (IMAGE) VALUES (?) ";
pStmt = conn.prepareStatement(sql);
pStmt.setBinaryStream(1, is, (int)(f.length()));
pStmt.execute();
conn.commit();
```

This will upload multipart file to your data base.

Note : get cos.jar from oreilly website

Question No : 2
How to Upload Image into Data Base ?
Answer :
Image upload from JSP to Data Base step by step :

step 1 :
In the JSP

```
<form name="regform2" method="post" enctype="multipart/form-data">
<input type="file" name="ImageFile" id="ImageFile" onChange="uploadImage()"/>
</form>
```

java script :

```
function uploadImage(){
document.regform.action ="<%=request.getContextPath()%>/dfdmin?cmd=uploadimage";
document.regform.submit();
}
```

Step 2.
In the servlet - add the below code.
This will upload your image to server (D:\\) or In unix you can mention /home/user like that

```
String rtempfile = File.createTempFile("temp","1").getParent();
MultipartRequest multi = new MultipartRequest(request, rtempfile, 500000 * 1024);
File rnewfile=null;
rnewfile = new
File("D:\\"+"jsp"+File.separator+"images"+File.separator+"uploadImage"+File.separator);
if(rnewfile.exists()){
}else{
rnewfile.mkdirs();
}

File f = multi.getFile("ImageFile");
System.out.println(f.getName());
FileInputStream fin =new FileInputStream(f);
RandomAccessFile r = new RandomAccessFile(rnewfile+File.separator+f.getName(),"rw");
filename = f.getName();
// FileOutputStream fos =new FileOutputStream(rnewfile);
byte sizefile[] = new byte[5000000];
fin.read(sizefile);
// fos.write(sizefile);
r.write(sizefile);
```

```
//fos.close();
r.close();
fin.close();
```

DAO.upload(f); // Call to DAO for insert image into database.

Step 3.

Then call to DAO to save this image to data base.
Insert into Database

```
public void upload(File f){
InputStream is = new FileInputStream(f);
String sql = " INSERT INTO image_upload (IMAGE) VALUES (?) ";
pStmt = conn.prepareStatement(sql);
pStmt.setBinaryStream(1, is, (int)(f.length()));
pStmt.execute();
conn.commit();
}
```

This will upload multipart file to your data base.

Note : get cos.jar from oreilly website

OR
if your server have already image file then just do below step

Code is here :

```
File f = new File("d:\\test.jpg");
InputStream is = new FileInputStream(f);
String sql = " INSERT INTO image_upload (IMAGE) VALUES (?) ";
pStmt = conn.prepareStatement(sql);
pStmt.setBinaryStream(1, is, (int)(f.length()));
 pStmt.execute();
conn.commit();
```

This will work.

Question No : 3

Q. What is Rowset ? CachedRowSet,JDBCRowSet and WebRowSet ?

Answer :

A RowSet object is a java bean component and extends the ResultSet interface.

The RowSet is majorly classified into two types :

1. Connected Rowset :

The connected rowset as the name suggests in connected to the database connection object like the resultset.

The JDBCRowSet is the example of the connected RowSet.

2. Disconnected RowSet:

The disconnected RowSet only connects to the database whenever required and after finishing the interaction they close the database connection.

There are three types of RowSet :

A CachedRowSet class?

a disconnected rowset that caches its data in memory; not suitable for very large data sets, but an ideal way to provide thin Java clients.

Example :

```
CachedRowSet cachedRs = new CachedRowSetImpl();
cachedRs.setUsername("scott");
cachedRs.setPassword("tiger");
cachedRs.setUrl("jdbc:oracle://localhost:3306/test");
cachedRs.setCommand("select * from employee");
cachedRs.setPageSize(4);
cachedRs.execute();
      while (cachedRs.nextPage()) {
  while (cachedRs.next()) {
          System.out.println(cachedRs.getString("emp_name"));
       }
  }
```

A JDBCRowSet class?

a connected rowset that serves mainly as a thin wrapper around a ResultSet object to make a JDBC driver look like a JavaBeans component.

Example:

```
JdbcRowSet jdbcRs = new JdbcRowSetImpl();
jdbcRs.setUsername("scott");
```

```
jdbcRs.setPassword("tiger");
jdbcRs.setUrl("jdbc:oracle://localhost:3306/test");
jdbcRs.setCommand("select * from employee");
jdbcRs.execute();
while(jdbcRs.next()) {
System.out.println(jdbcRs.getString("emp_name"));
  }
```

A WebRowSet class?
a connected rowset that uses the HTTP protocol internally to talk to a Java servlet that provides data access;
used to make it possible for thin web clients to retrieve and possibly update a set of rows.

Question No : 4
Q. What is Batch Updates Using Statements in JDBC ?
Answer :
Batch Updates calls to database as a chunk.
If you want to run more than one sql statement in a single database call then you have to go for Batch Update.
All Insert in single data base call and single transaction (if one fail the all fail)

For Example :

```
public class TestDB {
      public static void main(String[] args) {
          try {

            /** Loading the driver*/

                Class.forName("com.oracle.jdbc.Driver");

             /** Getting Connection*/
     Connection con =
DriverManager.getConnection("jdbc:oracle://localhost:3306/test","test","test");

            /** Creating Statement*/
              con.setAutoCommit(false); // This means after all insert  then commit at last
              Statement stmt = con.createStatement();
 stmt.addBatch("INSERT INTO EMP VALUES(1,'Ram1')");
 stmt.addBatch("INSERT INTO EMP VALUES(2,'Ram2')");
 stmt.addBatch("INSERT INTO EMP VALUES(3,'Ram3')");
```

```
  stmt.addBatch("INSERT INTO EMP VALUES(4,'Ram4')");
  int [] updateCounts= stmt.executeBatch();
  con.commit();
  con.setAutoCommit(true);
  /// All Insert in single data base call and single transaction ( if one fail the all fail)

                          stmt.close();
                               con.close();

                         } catch (Exception e) {

                               e.printStackTrace();
                         }
               }
                  }
```

Question No : 5

Q.Handling Blob and CLOB data using JDBC ? Insert CLOB and Retrive CLOB , Convert into String ?

Answer :

BLOB (Binary Large Objects) and CLOB(Character large objects) are special datatypes and can hold the large chunks of data in form of objects or text.

Blob and Clob objects persist the data of the objects into the database as a stream.

Example :

Java code to execute procedure

```
public class TestDB {
       public static void main(String[] args) {
          try {

            /** Loading the driver*/

                Class.forName("com.oracle.jdbc.Driver");

             /** Getting Connection*/
     Connection con =
DriverManager.getConnection("jdbc:oracle://localhost:3306/test","test","test");

           PreparedStatement pstmt = con.prepareStatement("insert into
Emp(id,name,description)values(?,?,?)");
```

```
            pstmt.setInt(1,5);
            pstmt.setString(2,"Das");
          // Create a big CLOB value...AND inserting as a CLOB
           StringBuffer sb = new StringBuffer(400000);

            sb.append("This is the  Example of CLOB ..");
            String clobValue = sb.toString();
            pstmt.setString(3, clobValue);
            int i= pstmt.executeUpdate();
            System.out.println("Done Inserted");
            pstmt.close();
     con.close();

     // Retrive CLOB values
 Connection con = DriverManager.getConnection("jdbc:oracle://localhost:3306/test","test","test");
  PreparedStatement pstmt = con.prepareStatement("select * from Emp where id=5");
  ResultSet rs = pstmt.executeQuery();
  Reader instream = null;
  int chunkSize;
  if(rs.next()){
  String name = rs.getString("name");
   java.sql.Clob  clob = result.getClob("description")
   StringBuffer sb1 = new StringBuffer();

   chunkSize = ((oracle.sql.CLOB)clob).getChunkSize();
   instream = clob.getCharacterStream();
   BufferedReader in = new BufferedReader(instream);
   String line = null;
   while ((line = in.readLine()) != null) {
   sb1.append(line);
 }
if(in != null){
in.close();
}

String clobdata = sb1.toString(); // this is the clob data converted into string

  }


                        } catch (Exception e) {
```

```
                    e.printStackTrace();
                }
            }
               }
```

Question No : 6

Q. Calling Stored procedures with Callable Statement ?

Answer :

Callable Statement is used to call stored procedure .

Step 1:

Write the procedure

```
CREATE OR REPLACE PROCEDURE getEmpAge
 (ID IN NUMBER, AGE OUT  NUMBER)

  IS
 Return_AGE NUMBER;
 BEGIN
  SELECT AGE INTO Return_AGE FROM EMP WHERE ID:=ID;
 Return_AGE;
 END;
```

Step 2:

Java code to execute procedure

```
public class TestDB {
        public static void main(String[] args) {
            try {

              /** Loading the driver*/

                  Class.forName("com.oracle.jdbc.Driver");

               /** Getting Connection*/
      Connection con =
DriverManager.getConnection("jdbc:oracle://localhost:3306/test","test","test");

              /** Creating CallableStatement*/

               CallableStatement call = con.prepareCall("call getEmpAge(?,?)");
         call.setInt(1,24); // setting emp id=24
```

```
            call.registerOutParameter(2,Types.INTEGER);
            call.execute();
             System.out.println("Emp Age is  "+call.getInt(2));


                                stmt.close();
                                con.close();

                        } catch (Exception e) {

                              e.printStackTrace();
                        }
             }
                }
```

Question No : 7

Q.What is JDBC ? Explain Types Of Drivers with Advantage and Disadvantage ?

Answer :

Java Database Connectivity(JDBC) defines how a java program can communicate with a database. JDBC API has two major packages java.sql and javax.sql.

There are 4 types of JDBC drivers available.

1) Type 1 Driver- the JDBC-ODBC bridge :

The JDBC type 1 driver, also known as the JDBC-ODBC bridge is a database driver implementation that the ODBC driver to connect to the database. The driver converts JDBC method calls into ODBC function calls. The bridge is usually used when there is no pure-Java driver available for a particular database.

The driver is implemented in the sun.jdbc.odbc.JdbcOdbcDriver class .

The driver is platform-dependent as it makes use of ODBC which in turn depends on native libraries of the operating system.

Advantage :

Almost any database for which ODBC driver is installed, can be accessed.

Disadvantage :

a) Performance overhead since the calls have to go through the JDBC overhead bridge to the ODBC driver.

b) The ODBC driver needs to be installed on the client machine

c) considering the client-side software needed, this might not be suitable for applets.

2) Type 2 Driver - the Native-API Driver :

The JDBC type 2 driver, also known as the Native-API driver is a database driver implementation that uses the client-side libraries of the database. The driver converts JDBC method calls into native calls of the database API.

The type 2 driver is not written entirely in Java as it interfaces with non-Java code that makes the final database calls.

A native-API partly Java technology-enabled driver converts JDBC calls into calls on the client API for ORACLE, DB2 or other . Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

However the type 2 driver provides more functionality and performance than the type 1 driver as it does not have the overhead of the additional ODBC function calls.

Advantage:
Better performance than Type 1 since no jdbc to odbc translation is needed

Disadvantage :
a) The vendor client library needs to be installed on the client machine.
b) Cannot be used in internet due the client side software needed
c) Not all databases give the client side library

3) Type 3 driver - the Network-Protocol Driver:
The JDBC type 3 driver, also known as the network-protocol driver is a database driver implementation which makes use of a middle-tier between the calling program and the database. The middle-tier (application server) converts JDBC calls directly or indirectly into the vendor-specific database protocol.

Advantages:
a) Since the communication between client and the middleware server is database independent, there is no need for the vendor db library on the client machine.
b) The Middleware Server (Can be a full fledged J2EE Application server) can provide typical middleware services like caching (connections, query results, and so on), load balancing etc.
Disadvantages :
a) Requires database-specific coding to be done in the middle tier.
b) An extra layer added may result in a time-bottleneck

4) Type 4 - the Native-Protocol Driver :
The JDBC type 4 driver, also known as the native-protocol driver is a database driver implementation that converts JDBC calls directly into the vendor-specific database protocol.
The type 4 driver is written completely in Java and is hence platform independent. It is installed inside the Java Virtual Machine of the client. It provides better performance over the type 1 and 2 drivers as it does not have the overhead of conversion of calls into ODBC or database API calls. Unlike the type 1 and 2 drivers, it does not need associated software to work..

Advantages :

a) These drivers don't translate the requests into db request to ODBC or pass it to client api for the db, nor do they need a middleware layer for request indirection. Thus the performance is considerably improved.
b) Web application mainly used this driver.
Disadvantage:
At client side, a separate driver is needed for each database. ex- classes12.zip (for ORACLE)

Question No : 8
Q. How java retrive the result returns from stored procedure ?
Answer :
In the Data Base :

```
create procedure getEmpId (VARCHAR) returns integer '
declare
   e_id NUMBER;
begin
    select emp_id into e_id from employee where name = $1;
return e_id;

end;
```

In the Java Code :

```
CallableStatement proc =
   connection.prepareCall("{ ? = call getEmpId (?) }");
proc.registerOutParameter(1, Types.INTEGER);
proc.setString(2, name);
cs.execute();
int empId = proc.getInt(2);
```

Question No : 9
Q.How to retrive the results in java returned by refcursor in Stored procedure ?
Answer :
In the Stored procedure :

```
create procedure emp_details () return refcursor as '
declare
   empref refcursor;
begin
   open empref for
      SELECT emp.name,emp.age
```

```
        FROM emp;
    return empref;
end;
```

In the Java Code :

```
Connection con = null;
    CallableStatement cstmt = null;
    try
    {
        con = ConnectionPool.getConnection();

         con.setAutoCommit(false);

        // Setup the call.
        cstmt = connection.prepareCall("{ ? = call emp_details () }");
// you can use
cstmt = connection.prepareCall("{call emp_details (?) }");

        cstmt.registerOutParameter(1, OracleTypes.CURSOR);
        cstmt.execute();

        ResultSet rs = (ResultSet)cstmt.getObject(1);
        while (rs.next())
        {
            String name = rs.getString(1);
            int age = rs.getInt(2);
        }
        rs.close();
    }
    catch (SQLException e)
    {
        cstmt.close();
        con.close();
    }
```

Question No : 10

Q.Reading an Oracle ARRAY from a stored procedure as an out ?

Answer :

ORCALE code :

Step 1. Create a Object in Oracle

```
CREATE OR REPLACE
TYPE      test.EMP_TYPE IS OBJECT (
   join_date DATE,
   emp_name           VARCHAR2(200)
)
```

Step 2. Create ORACLE ARRAY

```
CREATE OR REPLACE
TYPE      test.EMP_ARRAY AS VARYING ARRAY (5000) OF EMP_TYPE;
```

Step 3.

Create Stored procedure

```
CREATE OR REPLACE PACKAGE test.EMP_TEST AS
PROCEDURE EMP_RET (p_val   IN OUT NOCOPY EMP_ARRAY
}
// So what ever you want and store the data into the Array.
```

Java Code :

```
OracleCallableStatement ocs = null;
ResultSet rs = null;
 //Prepare statement and array
   ocs = (OracleCallableStatement) conn.prepareCall({?CALL  EMP_TEST. EMP_RET(?)?);
   ArrayDescriptor ad = null;
   ad = new ArrayDescriptor(?EMP_ARRAY?, conn);
ARRAY emp_aa = new ARRAY(ad, conn, null);
   ocs.setArray(1, emp_aaa);
   ocs.registerOutParameter(1,OracleTypes.ARRAY,?EMP_ARRAY?);
ocs.execute();
ARRAY array = (ARRAY) ocs.getArray(1);
if (array != null && array.length() > 0) {
     rs = array.getResultSet();
   }
while (rs.next()) {
        STRUCT rowStruct = (STRUCT) rs.getObject(1);
        Object[] cols = rowStruct.getAttributes();
        System.out.println(cols[1]);//for join_date
System.out.println(cols[2]); // for emp_name
}
```

Question No : 11

What is Metadata and why should I use it?

Answer :

Metadata (?data about data?) is information about one of two things: Database information

(java.sql.DatabaseMetaData), or Information about a specific ResultSet (java.sql.ResultSetMetaData). Use DatabaseMetaData to find information about your database, such as its capabilities and structure. Use ResultSetMetaData to find information about the results of an SQL query, such as size and types of columns

---

Question No : 12

How Class.forName() load the Driver and DriverManager.getConnection() return connection?

Answer :

These are the steps happing inside

Step 1.

Class.forName("com.mysql.jdbc.Driver") load the Driver.

Step 2.

In the com.mysql.jdbc.Driver class there is static bock .

That will execute because of static bock.

```
static
   {
      try
      {
         DriverManager.registerDriver(new Driver());
      }
      catch(SQLException E)
      {
         throw new RuntimeException("Can't register driver!");
      }
   }
```

This static block call DriverManager.registerDriver(new Driver());

Step 3.

Inside DriverManager.registerDriver() method.

```
public static synchronized void registerDriver(java.sql.Driver driver)
 throws SQLException {

 DriverInfo di = new DriverInfo();
 di.driver = driver;
 di.driverClass = driver.getClass();
 di.driverClassName = di.driverClass.getName();
 drivers.addElement(di);
 println("registerDriver: " + di);
   }
```

DriverManager class create a Vector name "drivers" and add the Driver class to the vector. Now In the Vector we have com.mysql.jdbc.Driver object.

Step 4.
To get connection we can
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/testDB","username","password");
In this method , it search for the Driver in the vector , if available then connect and return connection.

```
for (int i = 0; i < drivers.size(); i++) {
    DriverInfo di = (DriverInfo)drivers.elementAt(i);
// if "jdbc:mysql" keywork which is input , match with the driver in the vector  then {
Connection result = di.driver.connect(url, info);

}
}
```

Return connection;
This class getting jdbc:mysql://localhost:3306/testDB as input . This class check for the driver based on "jdbc:mysql" in this case and connect the driver.

Question No : 13
What will Class.forName do while loading drivers?
Answer :
These are the steps happing inside
Step 1.
Class.forName("com.mysql.jdbc.Driver") load the Driver.
Step 2.
In the com.mysql.jdbc.Driver class there is static bock . That will execute because of static bock.

```
static
   {
      try
      {
         DriverManager.registerDriver(new Driver());
      }
      catch(SQLException E)
      {
         throw new RuntimeException("Can't register driver!");
      }
   }
```

This static block call DriverManager.registerDriver(new Driver());

Step 3.
Inside DriverManager.registerDriver() method.

```
public static synchronized void registerDriver(java.sql.Driver driver)
 throws SQLException {

 DriverInfo di = new DriverInfo();
 di.driver = driver;
 di.driverClass = driver.getClass();
 di.driverClassName = di.driverClass.getName();
 drivers.addElement(di);
 println("registerDriver: " + di);
   }
```

DriverManager class create a Vector name drivers and add the Driver class to the vector.
Now In the Vector we have com.mysql.jdbc.Driver object.

Step 4.
To get connection we can

```
Connection con =
DriverManager.getConnection(?jdbc:mysql://localhost:3306/testDB?,?username?,?password?);
```

In this method , it search the Driver in the vector , if available then connect and return connection.

```
for (int i = 0; i < drivers.size(); i++) {
    DriverInfo di = (DriverInfo)drivers.elementAt(i);
// if ?jdbc:mysql? keywork withic in input , match with the driver in the vector {
Connection result = di.driver.connect(url, info);

}
}
```

Return connection;
This class getting jdbc:mysql://localhost:3306/testDB as input . This class check for the driver based on ?jdbc:mysql? in this case and connect the driver.

Question No : 14
How to Determining If a Result Set Is Scrollable?
Answer :
resultSet.getType() will return you the type.

```
try {
       // Get type of the result set
       int type = resultSet.getType();

       if (type == ResultSet.TYPE_SCROLL_INSENSITIVE
```

```
            || type == ResultSet.TYPE_SCROLL_SENSITIVE) {
            // Result set is scrollable
        } else {
            // Result set is not scrollable
        }
    } catch (SQLException e) {
    }
```

Question No : 15

How Getting the Number of Rows in a Table Using a Scrollable Result Set?

Answer :

This example gets the number of rows in a scrollable result set by moving the cursor to the last row of the result set and then calling ResultSet.getRow().

```
    try {
        // Create a scrollable result set
        Statement stmt = connection.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
        ResultSet resultSet = stmt.executeQuery("SELECT * FROM my_table");

        // Move to the end of the result set
        resultSet.last();

        // Get the row number of the last row which is also the row count
        int rowCount = resultSet.getRow();
    } catch (SQLException e) {
    }
```

Question No : 16

What is Scrollable Result Set ?

Answer :

scrollable result set allows the cursor to be moved to any row in the result set. Back and forward.

```
try {
        // Create an insensitive scrollable result set
        Statement stmt = connection.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);

        // Create a sensitive scrollable result set
        stmt = connection.createStatement(
```

```
        ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
    } catch (SQLException e) {
    }
```

Question No : 17
What does setAutoCommit do?

Answer :
When a connection is created, it is in auto-commit mode. This means that each individual SQL statement is treated as a transaction and will be automatically committed right after it is executed. The way to allow two or more statements to be grouped into a transaction is to disable auto-commit mode:

```
con.setAutoCommit(false);
```

Once auto-commit mode is disabled, no SQL statements will be committed until you call the method commit explicitly.

```
con.setAutoCommit(false);
PreparedStatement updateSales =
 con.prepareStatement( "UPDATE COFFEES SET SALES = ? WHERE COF_NAME LIKE ?");
updateSales.setInt(1, 50); updateSales.setString(2, "Colombian");
updateSales.executeUpdate();
PreparedStatement updateTotal =
 con.prepareStatement("UPDATE COFFEES SET TOTAL = TOTAL + ? WHERE COF_NAME
LIKE ?");
updateTotal.setInt(1, 50);
updateTotal.setString(2, "Colombian");
updateTotal.executeUpdate();
con.commit();
con.setAutoCommit(true);
```

Question No : 18
What is the difference between TYPE_SCROLL_INSENSITIVE , and TYPE_SCROLL_SENSITIVE?

Answer :
You will get a scrollable ResultSet object if you specify one of these ResultSet constants.The difference between the two has to do with whether a result set reflects changes that are made to it while it is open and whether certain methods can be called to detect these changes. Generally

speaking, a result set that is TYPE_SCROLL_INSENSITIVE does not reflect changes made while it is still open and one that is TYPE_SCROLL_SENSITIVE does. All three types of result sets will make changes visible if they are closed and then reopened:

```
Statement stmt =
 con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
ResultSet srs =
 stmt.executeQuery("SELECT COF_NAME, PRICE FROM COFFEES");
srs.afterLast();
while (srs.previous())
{
 String name = srs.getString("COF_NAME");
 float price = srs.getFloat("PRICE");
 System.out.println(name + " " + price);
}
```

Question No : 19
How to Make Updates to Updatable Result Sets?

Answer :
Another new feature in the JDBC 2.0 API is the ability to update rows in a result set using methods in the Java programming language rather than having to send an SQL command. But before you can take advantage of this capability, you need to create a ResultSet object that is updatable. In order to do this, you supply the ResultSet constant CONCUR_UPDATABLE to the createStatement method.

```
Connection con =
 DriverManager.getConnection("jdbc:mySubprotocol:mySubName");
Statement stmt =
 con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
ResultSet uprs =
 stmt.executeQuery("SELECT NAME, PRICE FROM COFFEES");
```

Question No : 20
How can you move the cursor in scrollable result sets?

Answer :
One of the new features in the JDBC 2.0 API is the ability to move a result set?s cursor backward

as well as forward. There are also methods that let you move the cursor to a particular row and check the position of the cursor.

Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_READ_ONLY);
ResultSet srs = stmt.executeQuery(?SELECT COF_NAME, PRICE FROM COFFEES?);

The first argument is one of three constants added to the ResultSet API to indicate the type of a ResultSet object: TYPE_FORWARD_ONLY, TYPE_SCROLL_INSENSITIVE , and TYPE_SCROLL_SENSITIVE. The second argument is one of two ResultSet constants for specifying whether a result set is read-only or updatable: CONCUR_READ_ONLY and CONCUR_UPDATABLE. The point to remember here is that if you specify a type, you must also specify whether it is read-only or updatable. Also, you must specify the type first, and because both parameters are of type int , the compiler will not complain if you switch the order. Specifying the constant TYPE_FORWARD_ONLY creates a nonscrollable result set, that is, one in which the cursor moves only forward. If you do not specify any constants for the type and updatability of a ResultSet object, you will automatically get one that is TYPE_FORWARD_ONLY and CONCUR_READ_ONLY.

Question No : 21
What are the different types of Statements with Example?

Answer :
Regular statement (use createStatement method), prepared statement (use prepareStatement method) and callable statement (use prepareCall)

statement :

Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT NAME, PRICE FROM COFFEES");
String s = rs.getString("NAME");

prepareStatement : precompiled statement
String sql = " select FIRST_NAME, LAST_NAME, USER_DATA where user_id = ? ";
prepareStatement pStmt = conn.prepareStatement(sql);
pStmt.setLong(1, user.getUserId());
rs = pStmt.executeQuery();
String s = rs.getString("FIRST_NAME");

callable statement :

A CallableStatement object contains a call to a stored procedure.

```
CallableStatement cs = con.prepareCall("{call SHOW_SUPPLIERS}");
ResultSet rs = cs.executeQuery();
```

Question No : 22
How can you retrieve data from the ResultSet?

Answer :

```
conn =  getConnection();

    String sql = " select FIRST_NAME from USER_DATA where user_id = ? ";
    pStmt = conn.prepareStatement(sql);
    pStmt.setLong(1, user.getUserId());
while(rs.next()){
String fname=rs.getString("FIRST_NAME");
}
```

Question No : 23
How can you create JDBC statements and what are they?

Answer :
A Statement object is what sends your SQL statement to the DBMS. You simply create a Statement object and then execute it, supplying the appropriate execute method with the SQL statement you want to send. For a SELECT statement, the method to use is executeQuery. For statements that create or modify tables, the method to use is executeUpdate. It takes an instance of an active connection to create a Statement object. In the following example, we use our Connection object con to create the Statement object

Example:

```
Statement stmt = con.createStatement("select * from emp");
rs = stmt.executeQuery();
```

Question No : 24

How can you make the connection?
Answer :
While making a JDBC connection we go through the following steps :

Step 1 : Register the database driver by using :

Class.forName(\" driver classs for that specific database\");

Class.forName("com.mysql.jdbc.Driver");

Step 2 : Now create a database connection using :

Connection con = DriverManager.getConnection(url,username,password);

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/art1",art1,art1);

Step 3: Now Create a query using :

Statement stmt = Connection.Statement(\"select EMP_NAME from EMP\");

Step 4 : Exceute the query :

ResultSet rs = stmt.exceuteQuery();

while(rs.next()){

System.out.println(rs.getString(EMP_NAME));

}

Question No : 25
What will Class.forName do while loading drivers?
Answer :
It is used to create an instance of a driver and register it with the
DriverManager. When you have loaded a driver, it is available for making a connection with a DBMS

Question No : 26
How can you load the drivers?

Answer :
Loading the driver or drivers you want to use is very simple and involves just one line of code. If, for example, you want to use the JDBC-ODBC Bridge driver, the following code will load it:

Class.forName(?sun.jdbc.odbc.JdbcOdbcDriver?);

Your driver documentation will give you the class name to use. For instance, if the class name is jdbc.DriverXYZ, you would load the driver with the following line of code:

Class.forName(?jdbc.DriverXYZ?);

Question No : 27
What is PreparedStatement ?
Answer :
PreparedStatement is precompiled statement. Fist time it compile the SQL query and next call it only pass the parameter value and execute, in the same connection.

PreparedStatement is not compiling query every time.
PreparedStatement is better for CLOB and BLOB object.
Example:
PreparedStatement pstmt = con.prepareStatement(
 "UPDATE table4 SET name = ? WHERE eid = ?");

pstmt.setString(1,"das");
pstmt.setInt(2,5);
pstmt.executeUpdate();

The object pstmt now contains the statement "UPDATE table4 SET m = ? WHERE x = ?", which

has already been sent to the DBMS and been prepared for execution.

Question No : 28
what is rowset? and how does it differ from resultset ?
Answer :
Both rowset and resultset is contains result data from executed query.
But after connection close resultset data will be lost. But in case of rowset you can get the data after connection close also.

Question No : 29
How to find total column from a resultset?
Answer :
int columnCnt = java.sql.ResultSetMetaData.getColumnCount();

Question No : 30
What is the fastest type of JDBC driver?
Answer :
Type 4 (JDBC Net pure Java Driver) is the fastest JDBC driver. Type 1 and Type 3 drivers will be slower than Type 2 drivers (the database calls are make at least three translations versus two), and Type 4 drivers are the fastest (only one translation).

Question No : 31
What are the different JDB drivers available?
Answer :
There are mainly four type of JDBC drivers available. They are:

Type 1 : JDBC-ODBC Bridge Driver - A JDBC-ODBC bridge provides JDBC API access via one or more ODBC drivers. Note that some ODBC native code and in many cases native database client code must be loaded on each client machine that uses this type of driver. Hence, this kind of driver is generally most appropriate when automatic installation and downloading of a Java technology application is not important. For information on the JDBC-ODBC bridge driver provided by Sun.

Type 2: Native API Partly Java Driver- A native-API partly Java technology-enabled driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

Type 3: Network protocol Driver- A net-protocol fully Java technology-enabled driver translates

JDBC API calls into a DBMS-independent net protocol which is then translated to a DBMS protocol by a server. This net server middleware is able to connect all of its Java technology-based clients to many different databases. The specific protocol used depends on the vendor. In general, this is the most flexible JDBC API alternative. It is likely that all vendors of this solution will provide products suitable for Intranet use. In order for these products to also support Internet access they must handle the additional requirements for security, access through firewalls, etc., that the Web imposes. Several vendors are adding JDBC technology-based drivers to their existing database middleware products.

Type 4: JDBC Net pure Java Driver - A native-protocol fully Java technology-enabled driver converts JDBC technology calls into the network protocol used by DBMSs directly. This allows a direct call from the client machine to the DBMS server and is a practical solution for Intranet access. Since many of these protocols are proprietary the database vendors themselves will be the primary source for this style of driver. Several database vendors have these in progress.

Question No : 32

Q.What are collection pools? What are the advantages?

Answer :

A connection pool create database connections on server start up and maintained in memory (vector). so that the connections may be reused.

If you need a connection the connection pool give you the connection from connection pool. So it perform better because no need to create connection to database. Create connection to Database is time consuming.

Example:

Genererally most of the application server has their cown connection pool..like weblogic, websphare , ATG, JBOSS.

just we need to set in properties file or xml file

min=10;

max=20;

Question No : 33

Q.What are different types of Transaction Isolation Levels?

Answer :

The isolation level describes the degree to which the data being updated is visible to other transactions. This is important when two transactions are trying to read the same row of a table. Imagine two transactions: A and B. Here three types of inconsistencies can occur:

Dirty-read: A has changed a row, but has not committed the changes. B reads the uncommitted data but his view of the data may be wrong if A rolls back his changes and updates his own

changes to the database.

Non-repeatable read: B performs a read, but A modifies or deletes that data later. If B reads the same row again, he will get different data.

Phantoms: A does a query on a set of rows to perform an operation. B modifies the table such that a query of A would have given a different result. The table may be inconsistent.

TRANSACTION_READ_UNCOMMITTED : DIRTY READS, NON-REPEATABLE READ AND PHANTOMS CAN OCCUR.

TRANSACTION_READ_COMMITTED : DIRTY READS ARE PREVENTED, NON-REPEATABLE READ AND PHANTOMS CAN OCCUR.
TRANSACTION_REPEATABLE_READ : DIRTY READS , NON-REPEATABLE READ ARE PREVENTED AND PHANTOMS CAN OCCUR.
TRANSACTION_SERIALIZABLE : DIRTY READS, NON-REPEATABLE READ AND PHANTOMS ARE PREVENTED.

Question No : 34
Q. What is Dirty read?
Answer :
A has changed a row, but has not committed the changes. B reads the uncommitted data but his view of the data may be wrong if A rolls back his changes and updates his own changes to the database.

Question No : 35
Q.Difference between Statement , PreparedStatement and CallableStatement ?
Answer :
Statement : Statement every time compile the SQL and Execute.
Example: String sql="select * from emp where emp_id = 1";
Statement stmt = conn.createStatement(sql);

PreparedStatement :
If we are using PreparedStatement the execution time will be less. First time RDBMS comiple the SQL and PreparedStatement is executed then other calls doesn't compile the SQL only execute the SQL within the connection live.

you must use a PreparedStatement object if you want to use large objects like BLOBs or CLOBs.

PreparedStatement is its support for batching

Example :
String sql="select * from emp where emp_id = ?";
PreparedStatement pStmt = conn.prepareStatement(sql);
pStmt.setLong(1, profile.getUserId());

CallableStatement :CallableStatement is for call to a stored procedure.
Example :
CallableStatement cs =
con.prepareCall("{call SHOW_CUSTOMER}");
ResultSet rs = cs.executeQuery();

Question No : 36
Q.What is the advantage of using PreparedStatement?
Answer :
If we are using PreparedStatement the execution time will be less. First time RDBMS comiple the SQL and PreparedStatement is executed then other calls doesn't compile the SQL only execute the SQL within the connection live.

you must use a PreparedStatement object if you want to use large objects like BLOBs or CLOBs.

PreparedStatement is its support for batching.

Question No : 37
Q.What is Single-Phase Commit ?
Answer :
If only a single resource (database) is enlisted in the transaction, you can use single-phase commit.
Example:
conn.setAutoCommit(false);
//do whatever
conn.commit();

Question No : 38

Q.What is two-phase commit?
Answer :
When a transaction involves multiple distributed resources, for example, a database server on each of two different network hosts, the commit process is somewhat complex because the transaction includes operations that span two distinct software systems, each with its own resource manager, log records, and so on. (In this case, the distributed resources are the database servers.)

Two-phase commit is a transaction protocol designed for the complications that arise with distributed resource managers. With a two-phase commit protocol, the distributed transaction manager employs a coordinator to manage the individual resource managers.

Configuring Two-Phase Commit Engine:
When a global transaction involves multiple databases, the changes to these resources must all be committed or rolled back at the same time. That is, when the transaction ends, the transaction manager contacts a coordinator--also known as a two-phase commit engine--to either commit or roll back all changes to all included databases. The two-phase commit engine is an Oracle9i database that is configured with the following:

Fully-qualified database links from itself to each of the databases involved in the transaction. When the transaction ends, the two-phase commit engine communicates with the included databases over their fully-qualified database links.

A user that is designated to create sessions to each database involved and is given the responsibility of performing the commit or rollback. The user that performs the communication must be created on all involved databases and be given the appropriate privileges.

Question No : 39
Q. How JDBC work with REF CURSOR returned by stored procedure and retrive results?
Answer :
REF CUSROR is retrieved by the JDBC program
into a ResultSet.
Example code : This will help you.
Using regular CallableStatement :

```
 CallableStatement cstmt = null;
 ResultSet         rset  = null;
cstmt = con.prepareCall(
              "{? = call ref_cursor_package.get_ref_cursor(?)}"
           );
           cstmt.registerOutParameter(1, OracleTypes.CURSOR);
```

```
        cstmt.setInt(2, 104);
        cstmt.execute();

        rset = (ResultSet) cstmt.getObject(1);

        while (rset.next()) {
           System.out.println(
              " - " +
              rset.getString(2) + " (" + rset.getInt(1) + "), " +
              rset.getString(3)
           );
        }
```

Using OracleCallableStatement and OracleResultSet classes:

```
OracleCallableStatement oraCallStmt   = null;
OracleResultSet         deptResultSet = null;
oraCallStmt = (OracleCallableStatement) con.prepareCall(
              "{? = call ref_cursor_package.get_dept_ref_cursor(?)}"
        );
        oraCallStmt.registerOutParameter(1, OracleTypes.CURSOR);
        oraCallStmt.setInt(2, 104);
        oraCallStmt.execute();

        deptResultSet = (OracleResultSet) oraCallStmt.getCursor(1);

        while (deptResultSet.next()) {
           System.out.println(
              " - " +
              deptResultSet.getString(2) + " (" + deptResultSet.getInt(1) + "), " +
              deptResultSet.getString(3)
           );
        }
```

Question No : 40

Q. How do i get result return from stored procedure in JDBC?

Answer :

```
CallableStatement cstmt = con.prepareCall("{ ? = call GetSeqNexVal(?) }");
    cstmt.registerOutParameter(1, java.sql.Types.INTEGER);
cstmt.setString(2,seq);
 cstmt.execute();
```

```
 id = cstmt.getInt(1);
System.out.println("Result Return from stored procedure"+id);
```

Question No : 41

Q.How do you call a Stored Procedure from JDBC?

Answer :

CallableStatement object is used to call stored procedure .

```
 CallableStatement cs =
  con.prepareCall("{call SHOW_CUSTOMER}");
 ResultSet rs = cs.executeQuery();
```

Question No : 42

Q.How do you handle your own transaction ?

Answer :

Connection Object has a method called setAutocommit(Boolean istrue)

- Default is true.

Set the Parameter to false , and begin your transaction

Example:

```
Connection conn =  getConnection();
conn.setAutocommit(false);
//other code
conn.commit();
conn.close();
```

Question No : 43

Q.What are the steps in the JDBC connection?

Answer :

While making a JDBC connection we go through the following steps :

Step 1 : Register the database driver by using :

```
Class.forName(\" driver classs for that specific database\" );
Class.forName("com.mysql.jdbc.Driver");
```

Step 2 : Now create a database connection using :

```
Connection con = DriverManager.getConnection(url,username,password);
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/art1",art1,art1);
```

Step 3: Now Create a query using :
Statement stmt = Connection.Statement(\"select EMP_NAME from EMP\");

Step 4 : Exceute the query :
ResultSet rs = stmt.exceuteQuery();
while(rs.next()){

System.out.println(rs.getString(EMP_NAME));

}

****

****
JMS Interview Questions
****
Question No : 1
Q. When to Remove messages from the queue ?
Answer :
When an application uses ConnectionConsumers, JMS might need to remove messages from the queue in a number of situations:
1) Badly formatted message

A message might arrive that JMS cannot parse.

2) Poison message
A message might reach the backout threshold, but the ConnectionConsumer fails to requeue it on the backout queue.

3)No interested ConnectionConsumer

For point-to-point messaging, when the QueueConnectionFactory is set so that it does not retain unwanted messages, a message arrives that is unwanted by any of the ConnectionConsumers.

Question No : 2
Q. What is poison messages? And how to handle poison messages?
Answer :
Poison messages, messages the application can never successfully process.
A badly-formatted message arrives on a queue. Such a message might make the receiving

application fail and back out the receipt of the message. In this situation, such a message might be received, then returned to the queue, repeatedly. These messages are known as poison messages. The ConnectionConsumer must be able to detect poison messages and reroute them to an alternative destination.

Question No : 3
What are the Message Headers in JMS message ?
Answer :
JMSDestination - send or publish method
JMSDeliveryMode - send or publish method
JMSExpiration - send or publish method
JMSPriority - send or publish method
JMSMessageID - send or publish method
JMSTimestamp - send or publish method
JMSCorrelationID - Client
JMSReplyTo - Client
JMSType - Client
JMSRedelivered - JMS provider

Question No : 4
What are the steps to send and receive JMS message ?
Answer :
Step 1.
lookup the ConnectionFactory .
(A connection factory is the object a client uses to create a connection to a provider. A connection factory encapsulates a set of connection configuration parameters that has been defined by an administrator. Each connection factory is an instance of the ConnectionFactory, QueueConnectionFactory, or TopicConnectionFactory interface)

```
Context ctx = new InitialContext();
ConnectionFactory connectionFactory = (ConnectionFactory)
  ctx.lookup("jms/ConnectionFactory");
```

Step 2.
lookup the destination (A destination is the object a client uses to specify the target of messages it produces and the source of messages it consumes.)

```
Destination myDest = (Destination) ctx.lookup("jms/MyTopic");
```
or
```
Queue myQueue = (Queue) ctx.lookup("jms/MyQueue");
```

Step 3.
Create the Connection
Connection connection = connectionFactory.createConnection();

Step 4.
Session session = connection.createSession(false,
  Session.AUTO_ACKNOWLEDGE);

The first argument means that the session is not transacted; the second means that the session automatically acknowledges messages when they have been received successfully

Step 5.
Create Message Producers

You use a Session to create a MessageProducer for a destination. Here, the first example creates a producer for the destination myQueue, and the second for the destination myTopic:

MessageProducer producer = session.createProducer(myQueue);
or
MessageProducer producer = session.createProducer(myTopic);

Step 6.
Create message and send message

TextMessage message = session.createTextMessage();
message.setText("testmsg");
producer.send(message);

Step 7.
create Message Consumers and receive message
MessageConsumer consumer = session.createConsumer(myQueue);
or
MessageConsumer consumer = session.createConsumer(myTopic);

Message m = consumer.receive();
if (m instanceof TextMessage) {
  TextMessage message = (TextMessage) m;
  System.out.println("Reading message: " + message.getText());
} else {
  // Handle error

}

Question No : 5
What is Publish/Subscribe Messaging in JMS ?
Answer :
Pub/sub messaging has the following characteristics :

a) Each message can have multiple consumers.

b)Publishers and subscribers have a timing dependency. A client that subscribes to a topic can consume only messages published after the client has created a subscription, and the subscriber must continue to be active in order for it to consume messages.

Question No : 6
What Point-to-Point Messaging in JMS ?
Answer :
a) Each message has only one consumer.
b) A sender and a receiver of a message have no timing dependencies. The receiver can fetch the message whether or not it was running when the client sent the message.
c)The receiver acknowledges the successful processing of a message.

Question No : 7
What are the main component in JMS?
Answer :
1)A JMS provider is a messaging system that implements the JMS interfaces and provides administrative and control features.

2)JMS clients are the programs or components, written in the Java programming language, that produce and consume messages. Any J2EE application component can act as a JMS client.

3)Messages are the objects that communicate information between JMS clients.

4)Administered objects are preconfigured JMS objects created by an administrator for the use of clients. The two kinds of JMS administered objects are destinations and connection factories
---

Question No : 8
How Does the JMS Work with the J2EE ?
Answer :
Application clients, Enterprise JavaBeans (EJB) components, and web components can send or synchronously receive a JMS message. Application clients can in addition receive JMS messages asynchronously. (Applets, however, are not required to support the JMS API.)

Message-driven beans, which are a kind of enterprise bean, enable the asynchronous consumption of messages. A JMS provider can optionally implement concurrent processing of messages by message-driven beans.

Message send and receive operations can participate in distributed transactions, which allow JMS operations and database accesses to take place within a single transaction.

Question No : 9
What Is Messaging?

Answer :
Messaging is a method of communication between software components or applications. A messaging system is a peer-to-peer facility: A messaging client can send messages to, and receive messages from, any other client. Each client connects to a messaging agent that provides facilities for creating, sending, receiving, and reading messages.

Question No : 10
Q. What is the difference between pub/sub and Point to Point i.e. p2p. ?
Answer :
pub/sub ? Publish/Subscriber
Publish/Subscriber can have multiple Publisher who publish message and multiple Subscriber who subscribe the messages. This is like watching TV multiple channel and multiple users .
One message can have more then one subscriber.
All the publish messages have the header and subscriber consume the message based on the header.
In JMS we say Topic.
P2p ? Point to point
One message have only one subscriber. Multiple publisher can send multiple messages to the Queue but each message has only one subscriber.
In JMS we say Queue.

Question No : 11
What is the different between JMS and RPC?

Answer :
RPC : Remote procedure Call
In RPC the method invoker waits for the method to finish execution and return the control back to the invoker. Thus it is completely synchronous in nature.

Example : In a website click on any link (country list link) and then website jsp call to database and get all the country list and shows in the screen.

JMS : Java Messaging System
While in JMS the message sender just sends the message to the destination and continues it's own processing. The sender does not wait for the receiver to respond. This is asynchronous behavior.

Example : Send a postal letter in post office and doing other works. Not waiting for post to receive.

Question No : 12
What are the types of messaging?

Answer :
There are two TYPES of Messaging.
Synchronous Messaging: Synchronous messaging involves a client that waits for the server to respond to a message.
Example : Call to database and got data
Asynchronous Messaging: Asynchronous messaging involves a client that does not wait for a message from the server. An event is used to trigger a message from a server.

Example : JMS

Question No : 13
What are the various message types supported by JMS?

Answer :
Stream Messages = Group of Java Primitives
Map Messages = Name Value Pairs. Name being a string& Value being a java primitive
Text Messages = String messages (since being widely used a separate messaging Type has been supported)
Object Messages = Group of serialize able java object
Bytes Message = Stream of uninterrupted bytes

****

****
EJB Interview Questions
****
Question No : 1
what are Container-Managed Transactional attributes ?
Answer :
NotSupported

The bean is not involved in a transaction. If the bean invoker calls the bean while involved in a transaction, the invoker's transaction is suspended, the bean executes, and when the bean returns, the invoker's transaction is resumed.

Required

The bean must be involved in a transaction. If the invoker is involved in a transaction, the bean uses the invoker's transaction. If the invoker is not involved in a transaction, the container starts a new transaction for the bean.

Supports

Whatever transactional state that the invoker is involved in is used for the bean. If the invoker has begun a transaction, the invoker's transaction context is used by the bean. If the invoker is not involved in a transaction, neither is the bean.

RequiresNew

Whether or not the invoker is involved in a transaction, this bean starts a new transaction that exists only for itself. If the invoker calls while involved in a transaction, the invoker's transaction is suspended until the bean completes.

Mandatory

The invoker must be involved in a transaction before invoking this bean. The bean uses the

invoker's transaction context.

Never

The bean is not involved in a transaction. Furthermore, the invoker cannot be involved in a transaction when calling the bean. If the invoker is involved in a transaction, a RemoteException is thrown.

Question No : 2
What's difference between httpsession and EJB session bean ?
Answer :
A session in a Servlet, is maintained by the Servlet Container through the HttpSession object, that is acquired through the request object. You cannot really instantiate a new HttpSession object, and it doesn't contains any business logic, but is more of a place where to store objects.

A session in EJB is maintained using the SessionBeans. You design beans that can contain business logic, and that can be used by the clients. You have two different session beans: Stateful and Stateless. The first one is somehow connected with a single client. It maintains the state for that client, can be used only by that client and when the client "dies" then the session bean is "lost".

A Stateless Session Bean doesn't maintain any state and there is no guarantee that the same client will use the same stateless bean, even for two calls one after the other. The lifecycle of a Stateless Session EJB is slightly different from the one of a Stateful Session EJB. Is EJB Container's responsability to take care of knowing exactly how to track each session and redirect the request from a client to the correct instance of a Session Bean. The way this is done is vendor dependant, and is part of the contract.

Question No : 3
What are the Differences between EJB 3.0 and EJB 2.1?
Answer :
Differences are :
1)EJB 3.0 allows developers to program EJB components as ordinary Java objects with ordinary

Java business interfaces rather than as heavy weight components like EJB 2 (home,remote).

2)In EJB 3.0 you can use annotaion or deployment descriptors but in EJB 2 you have to use deployment descriptors.

3) EJB 3 introduced persistence API for database access. In EJB 2 you can use Entity bean.

4) EJB 3.0 is much faster the EJB 2.

Question No : 4
Q. what are Container-Managed Transactional arributes ?
Answer :
NotSupported
 The bean is not involved in a transaction. If the bean invoker calls the bean while involved in a transaction, the invoker's transaction is suspended, the bean executes, and when the bean returns, the invoker's transaction is resumed.

Required
 The bean must be involved in a transaction. If the invoker is involved in a transaction, the bean uses the invoker's transaction. If the invoker is not involved in a transaction, the container starts a new transaction for the bean.

Supports
 Whatever transactional state that the invoker is involved in is used for the bean. If the invoker has begun a transaction, the invoker's transaction context is used by the bean. If the invoker is not involved in a transaction, neither is the bean.

RequiresNew
 Whether or not the invoker is involved in a transaction, this bean starts a new transaction that exists only for itself. If the invoker calls while involved in a transaction, the invoker's transaction is suspended until the bean completes.

Mandatory
 The invoker must be involved in a transaction before invoking this bean. The bean uses the invoker's transaction context.

Never
 The bean is not involved in a transaction. Furthermore, the invoker cannot be involved in a transaction when calling the bean. If the invoker is involved in a transaction, a RemoteException is thrown.

Question No : 5

What is the default transaction attribute for an EJB?

Answer :

There is no default transaction attribute for an EJB.

Deployer must specify a value for the transaction attribute for those methods having container managed transaction.

In WebLogic, the default transaction attribute for EJB is SUPPORTS.

WAS6.0 its "Required".

Question No : 6

Difference between SessionBean remove() and EntityBean remove() method?

Answer :

SessionBean remove() : inform the container of your loss of interest in this bean. Container will remove the instance.

EntityBean remove() : delete an entity bean without first instantiating it. Delete the row of the table using mentioned primary key.

Question No : 7

Why do we have a remove method in both EJBHome and EJBObject?

Answer :

With the EJBHome version of the remove, you are able to delete an entity bean without first instantiating it (you can provide a PrimaryKey object as a parameter to the remove method). The home version only works for entity beans. On the other hand, the Remote interface version works on an entity bean that you have already instantiated. In addition, the remote version also works on session beans (stateless and stateful) to inform the container of your loss of interest in this bean.

Question No : 8

Is it possible to share an HttpSession between a JSP and EJB? What happens when I change a value in the HttpSession from inside an EJB?

Answer :

You can pass the HttpSession as parameter to an EJB method, only if all objects in session are serializable.This has to be consider as ?passed-by-value?, that means that it?s read-only in the EJB. If anything is altered from inside the EJB, it won?t be reflected back to the HttpSession of the

Servlet Container.The ?pass-by-reference? can be used between EJBs Remote Interfaces, as they are remote references. While it IS possible to pass an HttpSession as a parameter to an EJB object, it is considered to be ?bad practice (1)? in terms of object oriented design. This is because you are creating an unnecessary coupling between back-end objects (ejbs) and front-end objects (HttpSession). Create a higher-level of abstraction for your ejb?s api. Rather than passing the whole, fat, HttpSession (which carries with it a bunch of http semantics), create a class that acts as a value object (or structure) that holds all the data you need to pass back and forth between front-end/back-end

---

Question No : 9

What is the difference between a ?Coarse Grained? Entity Bean and a ?Fine Grained? Entity Bean?

Answer :

A ?fine grained? entity bean is pretty much directly mapped to one relational table, in third normal form. A ?coarse grained? entity bean is larger and more complex, either because its attributes include values or lists from other tables, or because it ?owns? one or more sets of dependent objects. Note that the coarse grained bean might be mapped to a single table or flat file, but that single table is going to be pretty ugly, with data copied from other tables, repeated field groups, columns that are dependent on non-key fields, etc. Fine grained entities are generally considered a liability in large systems because they will tend to increase the load on several of the EJB server?s subsystems (there will be more objects exported through the distribution layer, more objects participating in transactions, more skeletons in memory, more EJB Objects in memory, etc.)

Question No : 10

What are the Interfaces need to create to implement Session Bean with Exmaple?

Answer :

Session bean class (CartBean)
Home interface (CartHome)
Remote interface (Cart)

Session bean class (CartBean) :

```
public class CartBean implements SessionBean {

  String customerName;
  String customerId;
  Vector contents;

  public void ejbCreate(String person)
    throws CreateException {

    if (person == null) {
```

```
      throw new CreateException("Null person not allowed.");
    }
    else {
      customerName = person;
    }

    customerId = "0";
    contents = new Vector();
  }

  public void ejbCreate(String person, String id)
    throws CreateException {

    if (person == null) {
      throw new CreateException("Null person not allowed.");
    }
    else {
      customerName = person;
    }

    IdVerifier idChecker = new IdVerifier();
    if (idChecker.validate(id)) {
      customerId = id;
    }
    else {
      throw new CreateException("Invalid id: "+ id);
    }

    contents = new Vector();
  }

  public void addBook(String title) {
    contents.addElement(title);
  }

  public void removeBook(String title) throws BookException {

    boolean result = contents.removeElement(title);
    if (result == false) {
      throw new BookException(title + "not in cart.");
    }
  }
```

```
  public Vector getContents() {
    return contents;
  }

  public CartBean() {}
  public void ejbRemove() {}
  public void ejbActivate() {}
  public void ejbPassivate() {}
  public void setSessionContext(SessionContext sc) {}

}
```

Home Interface:

```
public interface CartHome extends EJBHome {
  Cart create(String person) throws
              RemoteException, CreateException;
  Cart create(String person, String id) throws
              RemoteException, CreateException;
}
```

The signatures of the ejbCreate and create methods are similar, but differ in important ways. The rules for defining the signatures of the create methods of a home interface follow.

The number and types of arguments in a create method must match those of its corresponding ejbCreate method.
The arguments and return type of the create method must be valid RMI types.
A create method returns the remote interface type of the enterprise bean. (But an ejbCreate method returns void.)
The throws clause of the create method must include the java.rmi.RemoteException and the javax.ejb.CreateException

Remote Interface :

```
public interface Cart extends EJBObject {

  public void addBook(String title) throws RemoteException;
  public void removeBook(String title) throws
                BookException, RemoteException;
  public Vector getContents() throws RemoteException;
}
```

The method definitions in a remote interface must follow these rules:

Each method in the remote interface must match a method implemented in the enterprise bean class.
The signatures of the methods in the remote interface must be identical to the signatures of the corresponding methods in the enterprise bean class.
The arguments and return values must be valid RMI types.
The throws clause must include the java.rmi.RemoteException

Question No : 11
What are the parameters must follow for Session Bean ?
Answer :
It implements the SessionBean interface.
The class is defined as public.
The class cannot be defined as abstract or final.
It implements one or more ejbCreate methods.
It implements the business methods.
It contains a public constructor with no parameters.
It must not define the finalize method.

Question No : 12
What are the callbacks method in Session Bean ?
Answer :

```
 public void ejbCreate() {}
public void ejbRemove() {}
   public void ejbActivate() {}
   public void ejbPassivate() {}
   public void setSessionContext(SessionContext sc) {}
```

Question No : 13
What are the ways for a client application to get an EJB object?

Answer :
1) The client has the JNDI name of the EJB object; this name is used to get the EJB object.

2)The client has the JNDI name of the Home object, this is a more usual case; this name is used to get the Home object, then a finder method is invoked on this Home to obtain one or several entity bean objects. The client may also invoke a "create" method on the Home object to create a

new EJB object (session or entity).

3)The client has got a handle on an EJB object. A handle is an object that identifies an EJB object; it may be serialized, stored, and used later by a client to obtain a reference to the EJB Object, using the getEJBObject method().

4)The client has got a reference to an EJB object, and some methods defined on the remote interface of this Enterprise Bean return EJB objects.

Question No : 14

What is handle and why it is used in EJB?

Answer :

The handle mechanism allows a client application to maintain a reference to an EJB object. A handle object may be obtained by calling the getHandle() method on the reference to an EJB object. The main interest is that the handle class implements java.io.serializable interface, which means that a handle may be serialized. This allows the client to store the handle, or to pass it to another process. The handle may then be deserialized and used to obtain the reference to the EJB object, by calling the getEJBObject() method.

Handles on session bean objects are valid until the session bean object exists, i.e. their life time is limited to that of the client. Handles on entity bean objects are valid during the complete life time of the entity bean object; this means that such handles may be used by different clients and stored for a long time; the EJB server holding the entity bean objects may be stopped and restarted, the handle will still be valid.

If we consider the entity bean object of the example above (a2), the way to obtain a handle on this object is the following (the handle class is defined in the javax.ejb package):

Handle h = a2.getHandle();The handle object may then be serialized and stored in a file:

```
ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("handlefile"));
out.writeObject(h);
out.close();
```

Then, a client can read the handle, and retrieve the referenced object:

```
ObjectInputStream in = new ObjectInputStream(new FileInputStream("handlefile"));
Handle h  = (Handle) in.readObject();
Account a = (Account)PortableRemoteObject.narrow(h.getEJBObject(),
                                    Account.class);
```

The EJB Specification allows the client to obtain a handle for the home interface. This allows the client to store a reference to an entity bean's home interface in stable storage. The client code

must use the javax.rmi.PortableRemoteObject.narrow(...) method to convert the result of the getEJBHome() method invoked on a handle to the home interface type

---
Question No : 15
What is an EJB Context?
Answer :
EJBContext is an interface that is implemented by the container, and it is also a part of the bean-container contract. Entity beans use a subclass of EJBContext called EntityContext. Session beans use a subclass called SessionContext. These EJBContext objects provide the bean class with information about its container, the client using the bean and the bean itself. They also provide other functions. See the API docs and the spec for more details
---
Question No : 16
Implement Local and Remote Interfaces in EJB?
Answer :
Remote BeansThe EJB 1.1 specification defines all EJBs as remote objects. This means that every time you make a call to an EJB, you are making a remote call. This means that there is considerable overhead to each EJB call, and hence performance implications. To combat this, server vendors invented a way of circumventing the remote calls to some degree. Oracle's solution with OC4J was the pass-by-reference setting, which determined whether EJB objects were communicated by reference to the object, or whether the whole object had to be passed to the client.

An EJB has a remote interface and a home interface, with the exception of MessageDrivenBeans. The remote interface extends the interface javax.ejb.EJBObject and the home interface extends the interface javax.ejb.EJBHome. The EJB is accessible from any client, in any JVM, provided they have the proper authorization.

For example, the Home and Remote interfaces of an EJB called EMP may look like this.

Remote:

```
public interface Emp extends EJBObject
{
   long getEmpno() throws RemoteException;
   void setEmpno(long newDeptno) throws RemoteException;
   String getEname() throws RemoteException;
   void setEname(String newDname) throws RemoteException;
```

Home:

```
public interface DeptHome extends EJBHome
{
   public Emp create() throws RemoteException, CreateException;
   public Dept findByPrimaryKey(DeptPK primaryKey) throws RemoteException, FinderException;
```

Note that both the home and the remote interface throw a RemoteException in all of their method definitions. The ejb-jar.xml deployment descriptor for these EJBs would look something like the snippets below:

```
<entity>
  <ejb-name>Emp</ejb-name>
  <home>ejb.cmplocal.EmpHome</home>
  <remote>ejb.cmplocal.Emp</remote>
  <ejb-class>ejb.cmplocal.EmpBean</ejb-class>
  .
  .
  .
```

Local BeansThe EJB 2.0 specification standardize a means of making local connections to EJBs with Local Interfaces.

For an EJB to be classed as a local EJB, it must implement the local versions of the home and remote interfaces, javax.ejb.EJBLocalObject for the Home interface, and javax.ejb.EJBLocalHome. For a client to call the Local interface, they must be running in the same JVM as the JVM that the EJB exists in. This means that not only an EJB can call a local EJB , Servlets or JSPs can also call the EJB via it's local interface if they are packaged together as part of same application.

For example, the LocalHome and Local interfaces of an EJB called EMP may look like this.

Local:

```
public interface Emp extends EJBLocalObject
{
   long getEmpno();
   void setEmpno(long newEmpno);
   String getEname();
   void setEname(String newEname);
```

LocalHome:

```
public interface EmpHome extends EJBLocalHome
{
```

```
    public Emp create() throws CreateException;
    public Emp findByPrimaryKey(EmpPK primaryKey) throws FinderException;
```

The ejb-jar.xml deployment descriptor for these EJBs would look something like the snippets below:

```
 <entity>
   <ejb-name>Emp</ejb-name>
   <local-home>ejb.cmplocal.EmpHome</local-home>
   <local>ejb.cmplocal.Emp</local>
   <ejb-class>ejb.cmplocal.EmpBean</ejb-class>
   <cmp-version>2.x</cmp-version>
   <abstract-schema-name>Emp</abstract-schema-name>
   .
   .
   .
```

Note that now the local interfaces no longer throw the RemoteException, showing that they are not remotely called methods. Also, the XML contains different elements. There is now a local-home and a local tag. Also we are declaring that this is an EJB 2.x bean, using the cmp-version tag.

Calling Local BeansCalling a local bean from Java code is very simple, and very similar to using a remote bean. The code to call a remote bean is shown below.

```
 try
 {
   Context ctx = new InitialContext();
   Object o = ctx.lookup("Emp");
   EmpHome empHome = PortableRemoteObject.narrow(o, EmpHome.class)
   return empHome.findByDeptno(getDeptno());
 }
 catch (RemoteException r)
 {
   System.err.println("Error loading Employees(Remote): " + r.getMessage()); return null;
 }
 catch (NamingException n)
 {
   System.err.println("Error loading Employees(Naming): " + n.getMessage());
   return null;
 }
 catch (FinderException f)
 {
   System.err.println("Error loading Employees(Finder): " + f.getMessage());
```

```
    return null;
  }
```
The code for a local bean is similar, but we no longer have to worry about the PortableRemoteObject, as the bean is no longer remote.

```
  try
  {
    Context ctx = new InitialContext();
    Object o = ctx.lookup("java:comp/env/LocalEmp");
    EmpHome empHome = (EmpHome)o;
    return empHome.findByDeptno(getDeptno());
  }
  catch (NamingException n)
  {
    System.err.println("Error loading Employees(Naming): " + n.getMessage());
    return null;
  }
  catch (FinderException f)
  {
    System.err.println("Error loading Employees(Finder): " + f.getMessage());
    return null;
  }
```

As you can see, the local bean has to lookup the EJB slightly differently, even though they are running in the same container. Also, there is no RemoteException thrown by the find or the create methods, so the exception does not have to be caught.

There is one more difference, and that is in the ejb-jar.xml deployment descriptor. For an EJB to look up a local EJB, it must point to the correct location using an <ejb-local-ref> tag. If this is not used, the container will not be able to find the bean. For each EJB that needs to use the local EJB, the XML below must be in the deployment descriptor.

```
  <entity>
    <ejb-name>Dept</ejb-name>
    .
    .
    .
    <ejb-local-ref>
      <ejb-ref-name>LocalEmp</ejb-ref-name>
      <ejb-ref-type>Entity</ejb-ref-type>
      <local-home>ejb.cmplocal.EmpHome</local-home>
      <local>ejb.cmplocal.Emp</local>
```

```
    <ejb-link>Emp</ejb-link>
   </ejb-local-ref>
  </entity>
```

This example will allow the EJB Dept to call the local EJB Emp using the name LocalEmp. This is required because EJBs can have both local and remote interfaces, and to call the EJB Emp via it's remote interface the EJB Dept would look up the name Emp rather than the local reference LocalHome.

Question No : 17

How can I call one EJB from inside of another EJB?

Answer :

In case of Remote :

EJBs can be clients of other EJBs. It just works. Really. Use JNDI to locate the Home Interface of the other bean, then acquire an instance reference.

For Example : Context ctx = new InitialContext();

```
//get Home interface of bean
//narrow -retype
EmpHome lhome = (EmpHome )
javax.rmi.PortableRemoteObject.narrow(ctx.lookup("java:comp/env/LocalEmp"), EmpHome
.class);
//get remote interface
Emplbean = lhome.create();
//now you can call bussiness method on remote interface like
lbean.doSomething()
```

Incase of Local : but we no longer have to worry about the PortableRemoteObject, as the bean is no longer remote

```
Context ctx = new InitialContext();
   Object o = ctx.lookup("java:comp/env/LocalEmp");
   EmpHome empHome = (EmpHome)o;
```

Question No : 18

What is the difference between Message Driven Beans and Stateless Session beans?

Answer :
In several ways, the dynamic creation and allocation of message-driven bean instances mimics the behavior of stateless session EJB instances, which exist only for the duration of a particular method call. However, message-driven beans are different from stateless session EJBs (and other types of EJBs) in several significant ways:

Message-driven beans process multiple JMS messages asynchronously, rather than processing a serialized sequence of method calls.

Message-driven beans have no home or remote interface, and therefore cannot be directly accessed by internal or external clients. Clients interact with message-driven beans only indirectly, by sending a message to a JMS Queue or Topic.

Note: Only the container directly interacts with a message-driven bean by creating bean instances and passing JMS messages to those instances as necessary.

The Container maintains the entire lifecycle of a message-driven bean; instances cannot be created or removed as a result of client requests or other API calls

---
Question No : 19
What happens if remove( ) is never invoked on a session bean?
Answer :
In case of a stateless session bean it may not matter if we call or not as in both cases nothing is done. The number of beans in cache is managed by the container.

In case of stateful session bean, the bean may be kept in cache till either the session times out, in which case the bean is removed or when there is a requirement for memory in which case the data is cached and the bean is sent to free pool.

Question No : 20
Can you control when passivation occurs?

Answer :
The developer, according to the specification, cannot directly control when passivation occurs. Although for Stateful Session Beans, the container cannot passivate an instance that is inside a transaction. So using transactions can be a a strategy to control passivation.

The ejbPassivate() method is called during passivation, so the developer has control over what to do during this exercise and can implement the require optimized logic.

Some EJB containers, such as BEA WebLogic, provide the ability to tune the container to minimize passivation calls.

Taken from the WebLogic 6.0 DTD -"The passivation-strategy can be either "default" or "transaction". With the default setting the container will attempt to keep a working set of beans in the cache. With the "transaction" setting, the container will passivate the bean after every transaction (or method call for a non-transactional invocation).

Question No : 21
Can the primary key in the entity bean be a Java primitive type such as int?

Answer :
The primary key can't be a primitive type--use the primitive wrapper classes, instead. For example, you can use java.lang.Integer as the primary key class, but not int (it has to be a class, not a primitive)

Question No : 22
The EJB container implements the EJBHome and EJBObject classes. For every request from a unique client, does the container create a separate instance of the generated EJBHome and EJBObject classes?

Answer :
The EJB container maintains an instance pool. The container uses these instances for the EJB Home reference irrespective of the client request. while refering the EJB Object classes the container creates a separate instance for each client request. The instance pool maintainence is up to the implementation of the container. If the container provides one, it is available otherwise it is not mandatory for the provider to implement it. Having said that, yes most of the container providers implement the pooling functionality to increase the performance of the application server. The way it is implemented is again up to the implementer.

Question No : 23
How can i maintain a user session between servlets and stateful session ejbs?
Answer :
HttpSession and EJB Stateful session is different. HttpSession is in servlet container and EJB Stateful session is in Application Server EJB Container.

You can maintain the user session between servlets and stateful session ejbs using Handle Object of EJB.

Step 1. get the Handle Object using your
EJBObject.getHandle() method.
Exmaple : Bean lbean = lhome.create();
then Handle hd = lbean.getHandle();
Step 2.
session.setAttribute("handle",hd);
You can have question like why we can't store the session bean itself ?
answer is : ejb session bean is very heavy .
Step 3.
When you want the EJBObject just
Handle hd = session.getAttribute("handle");
Bean lbean = (Bean)hd.getEJBObject();

Question No : 24
What's difference between Servlet/JSP session and EJB session
Answer :
A session in a Servlet, is maintained by the Servlet Container through the HttpSession object, that is acquired through the request object. You cannot really instantiate a new HttpSession object, and it doesn't contains any business logic, but is more of a place where to store objects.

A session in EJB is maintained using the SessionBeans. You design beans that can contain business logic, and that can be used by the clients. You have two different session beans: Stateful and Stateless. The first one is somehow connected with a single client. It maintains the state for that client, can be used only by that client and when the client "dies" then the session bean is "lost".

A Stateless Session Bean doesn't maintain any state and there is no guarantee that the same client will use the same stateless bean, even for two calls one after the other. The lifecycle of a Stateless Session EJB is slightly different from the one of a Stateful Session EJB. Is EJB Container's responsability to take care of knowing exactly how to track each session and redirect the request from a client to the correct instance of a Session Bean. The way this is done is vendor dependant, and is part of the contract.

Question No : 25
Is it possible to share an HttpSession between a JSP and EJB? What happens when I change a

value in the HttpSession from inside an EJB?

Answer :
You can pass the HttpSession as parameter to an EJB method, only if all objects in session are serializable.This has to be consider as ?passed-by-value", that means that it?s read-only in the EJB. If anything is altered from inside the EJB, it won?t be reflected back to the HttpSession of the Servlet Container.The ?pass-by-reference? can be used between EJBs Remote Interfaces, as they are remote references. While it IS possible to pass an HttpSession as a parameter to an EJB object, it is considered to be ?bad practice ? in terms of object oriented design. This is because you are creating an unnecessary coupling between back-end objects (ejbs) and front-end objects (HttpSession). Create a higher-level of abstraction for your ejb?s api. Rather than passing the whole, fat, HttpSession (which carries with it a bunch of http semantics), create a class that acts as a value object (or structure) that holds all the data you need to pass back and forth between front-end/back-end. Consider the case where your ejb needs to support a non-http-based client. This higher level of abstraction will be flexible enough to support it.

Question No : 26
How to call any EJB from a servlet/JSP/Java Client?
Answer :
Context ctx = new InitialContext();

//get Home interface of bean
//narrow -retype
BeanHome lhome = (BeanHome)
javax.rmi.PortableRemoteObject.narrow(ctx.lookup("cz.train.Bean"), BeanHome.class);

//get remote interface
Bean lbean = lhome.create();

//now you can call bussiness method on remote interface like
lbean.doSomething()

Question No : 27
What are transaction isolation levels in EJB?

Answer :
1. Transaction_read_uncommitted- Allows a method to read uncommitted data from a DB(fast but not wise).

2. Transaction_read_committed- Guarantees that the data you are getting has been committed.

3. Transaction_repeatable_read - Guarantees that all reads of the database will be the same during the transaction (good for read and update operations).

4. Transaction_serializable- All the transactions for resource are performed serial.

Question No : 28
What are transaction attributes?

Answer :
The transaction attribute specifies how the Container must manage transactions for a method when a client invokes the method via the enterprise bean?s home or component interface or when the method is invoked as the result of the arrival of a JMS message. (Sun's EJB Specification) Below is a list of transactional attributes:

1. NotSupported - transaction context is unspecified.

2. Required - bean's method invocation is made within a transactional context. If a client is not associated with a transaction, a new transaction is invoked automatically.

3. Supports - if a transactional context exists, a Container acts like the transaction attribute is Required, else - like NotSupported.

4. RequiresNew - a method is invoked in a new transaction context.

5. Mandatory - if a transactional context exists, a Container acts like the transaction attribute is Required, else it throws a javax.ejb.TransactionRequiredException.

6. Never - a method executes only if no transaction context is specified.

Question No : 29
What is bean managed transaction?

Answer :
If a developer doesn't want a Container to manage transactions, it's possible to implement all database operations manually by writing the appropriate JDBC code. This often leads to productivity increase, but it makes an Entity Bean incompatible with some databases and it

enlarges the amount of code to be written. All transaction management is explicitly performed by a developer.

Question No : 30
Can Entity Beans have no create() methods?

Answer :
Yes. In some cases the data is inserted NOT using Java application, so you may only need to retrieve the information, perform its processing, but not create your own information of this kind.

Question No : 31
What are the callback methods in Entity beans?

Answer :
The bean class defines create methods that match methods in the home interface and business methods that match methods in the remote interface. The bean class also implements a set of callback methods that allow the container to notify the bean of events in its life cycle. The callback methods are defined in the javax.ejb.EntityBean interface that is implemented by all entity beans.The EntityBean interface has the following definition. Notice that the bean class implements these methods.

```
public interface javax.ejb.EntityBean {
  public void setEntityContext();
public void unsetEntityContext();
public void ejbLoad();
public void ejbStore();
public void ejbActivate();
public void ejbPassivate();
public void ejbRemove();
}
```

The setEntityContext() method provides the bean with an interface to the container called the EntityContext. The EntityContext interface contains methods for obtaining information about the context under which the bean is operating at any particular moment. The EntityContext interface is used to access security information about the caller; to determine the status of the current transaction or to force a transaction rollback; or to get a reference to the bean itself, its home, or its primary key. The EntityContext is set only once in the life of an entity bean instance, so its reference should be put into one of the bean instance's fields if it will be needed later.

The unsetEntityContext() method is used at the end of the bean's life cycle before the instance is evicted from memory to dereference the EntityContext and perform any last-minute clean-up.

The ejbLoad() and ejbStore() methods in CMP entities are invoked when the entity bean's state is being synchronized with the database. The ejbLoad() is invoked just after the container has refreshed the bean container-managed fields with its state from the database. The ejbStore() method is invoked just before the container is about to write the bean container-managed fields to the database. These methods are used to modify data as it's being synchronized. This is common when the data stored in the database is different than the data used in the bean fields.

The ejbPassivate() and ejbActivate() methods are invoked on the bean by the container just before the bean is passivated and just after the bean is activated, respectively. Passivation in entity beans means that the bean instance is disassociated with its remote reference so that the container can evict it from memory or reuse it. It's a resource conservation measure the container employs to reduce the number of instances in memory. A bean might be passivated if it hasn't been used for a while or as a normal operation performed by the container to maximize reuse of resources. Some containers will evict beans from memory, while others will reuse instances for other more active remote references. The ejbPassivate() and ejbActivate() methods provide the bean with a notification as to when it's about to be passivated (disassociated with the remote reference) or activated (associated with a remote reference).

Question No : 32

What is the difference between Container-Managed Persistent (CMP) bean and Bean-Managed Persistent(BMP) ?

Answer :

Container-managed persistence(CMP) beans are the simplest for the bean developer to create and the most difficult for the EJB server to support. This is because all the logic for synchronizing the bean's state with the database is handled automatically by the container. This means that the bean developer doesn't need to write any data access logic, while the EJB server is supposed to take care of all the persistence needs automatically. With CMP, the container manages the persistence of the entity bean. A CMP bean developer doesn't need to worry about JDBC code and transactions, because the Container performs database calls and transaction management instead of the programmer. Vendor tools are used to map the entity fields to the database and absolutely no database access code is written in the bean class. All table mapping is specified in the deployment descriptor. Otherwise, a BMP bean developer takes the load of linking an application and a database on his shoulders.

The bean-managed persistence (BMP) enterprise bean manages synchronizing its state with the database as directed by the container. The bean uses a database API to read and write its fields

to the database, but the container tells it when to do each synchronization operation and manages the transactions for the bean automatically. Bean-managed persistence gives the bean developer the flexibility to perform persistence operations that are too complicated for the container or to use a data source that is not supported by the container.BMP beans are not 100% database-independent, because they may contain database-specific code, but CMP beans are unable to perform complicated DML (data manipulation language) statements. EJB 2.0 specification introduced some new ways of querying database (by using the EJB QL - query language).

Question No : 33

What are the methods of Entity Bean?

Answer :

An entity bean consists of 4 groups of methods:

1. create methods: To create a new instance of a CMP entity bean, and therefore insert data into the database, the create() method on the bean's home interface must be invoked. They look like this: EntityBeanClass ejbCreateXXX(parameters), where EntityBeanClass is an Entity Bean you are trying to instantiate, ejbCreateXXX(parameters) methods are used for creating Entity Bean instances according to the parameters specified and to some programmer-defined conditions.

A bean's home interface may declare zero or more create() methods, each of which must have corresponding ejbCreate() and ejbPostCreate() methods in the bean class. These creation methods are linked at run time, so that when a create() method is invoked on the home interface, the container delegates the invocation to the corresponding ejbCreate() and ejbPostCreate() methods on the bean class.

2. finder methods: The methods in the home interface that begin with "find" are called the find methods. These are used to query the EJB server for specific entity beans, based on the name of the method and arguments passed. Unfortunately, there is no standard query language defined for find methods, so each vendor will implement the find method differently. In CMP entity beans, the find methods are not implemented with matching methods in the bean class; containers implement them when the bean is deployed in a vendor specific manner. The deployer will use vendor specific tools to tell the container how a particular find method should behave. Some vendors will use object-relational mapping tools to define the behavior of a find method while others will simply require the deployer to enter the appropriate SQL command.

There are two basic kinds of find methods: single-entity and multi-entity. Single-entity find methods return a remote reference to the one specific entity bean that matches the find request. If no entity beans are found, the method throws an ObjectNotFoundException . Every entity bean must define the single-entity find method with the method name findByPrimaryKey(), which takes the bean's primary key type as an argument.

The multi-entity find methods return a collection (Enumeration or Collection type) of entities that match the find request. If no entities are found, the multi-entity find returns an empty collection.

3. remove methods: These methods (you may have up to 2 remove methods, or don't have them at all) allow the client to physically remove Entity beans by specifying either Handle or a Primary Key for the Entity Bean.

4. home methods: These methods are designed and implemented by a developer, and EJB specification doesn't have any requirements for them except the need to throw a RemoteException is each home method.

Question No : 34
What is Entity Bean?

Answer :
The entity bean is used to represent data in the database. It provides an object-oriented interface to data that would normally be accessed by the JDBC or some other back-end API. More than that, entity beans provide a component model that allows bean developers to focus their attention on the business logic of the bean, while the container takes care of managing persistence,transactions, and access control.

There are two basic kinds of entity beans: container-managed ersistence (CMP) andbean-managed persistence (BMP).

Container-managed persistence beans are the simplest for the bean developer to create and the most difficult for the EJB server to support. This is because all the logic for synchronizing the bean's state with the database is handled automatically by the container. This means that the bean developer doesn't need to write any data access logic, while the EJB server is supposed to take care of all the persistence needs automatically. With CMP, the container manages the persistence of the entity bean. Vendor tools are used to map the entity fields to the database and absolutely no database access code is written in the bean class.

The bean-managed persistence (BMP) enterprise bean manages synchronizing its state with the database as directed by the container. The bean uses a database API to read and write its fields to the database, but the container tells it when to do each synchronization operation and manages the transactions for the bean automatically. Bean-managed persistence gives the bean developer the flexibility to perform persistence operations that are too complicated for the container or to use a data source that is not supported by the container.

Question No : 35
What is Session Bean?

Answer :
A session bean is a non-persistent object that implements some business logic running on the server. One way to think of a session object is as a logical extension of the client program that runs on the server.

Session beans are used to manage the interactions of entity and other session beans,access resources, and generally perform tasks on behalf of the client.

There are two basic kinds of session bean: stateless and stateful.

Stateless session beans are made up of business methods that behave like procedures; they operate only on the arguments passed to them when they are invoked. Stateless beans are called stateless because they are transient; they do not maintain business state between method invocations.Each invocation of a stateless business method is independent from previous invocations. Because stateless session beans are stateless, they are easier for the EJB container to manage, so they tend to process requests faster and use less resources.

Stateful session beans encapsulate business logic and state specific to a client. Stateful beans are called "stateful" because they do maintain business state between method invocations, held in memory and not persistent. Unlike stateless session beans, clients do not share stateful beans. When a client creates a stateful bean, that bean instance is dedicated to service only that client. This makes it possible to maintain conversational state, which is business state that can be shared by methods in the same stateful bean.

Question No : 36
What are the different kinds of enterprise beans?

Answer :
Stateless session bean- An instance of these non-persistent EJBs provides a service without storing an interaction or conversation state between methods. Any instance can be used for any client.

Stateful session bean- An instance of these non-persistent EJBs maintains state across methods and transactions. Each instance is associated with a particular client.

Entity bean- An instance of these persistent EJBs represents an object view of the data, usually rows in a database. They have a primary key as a unique identifier. Entity bean persistence can be either container-managed or bean-managed.

Message-driven bean- An instance of these EJBs is integrated with the Java Message Service (JMS) to provide the ability for message-driven beans to act as a standard JMS message consumer and perform asynchronous processing between the server and the JMS message

producer.

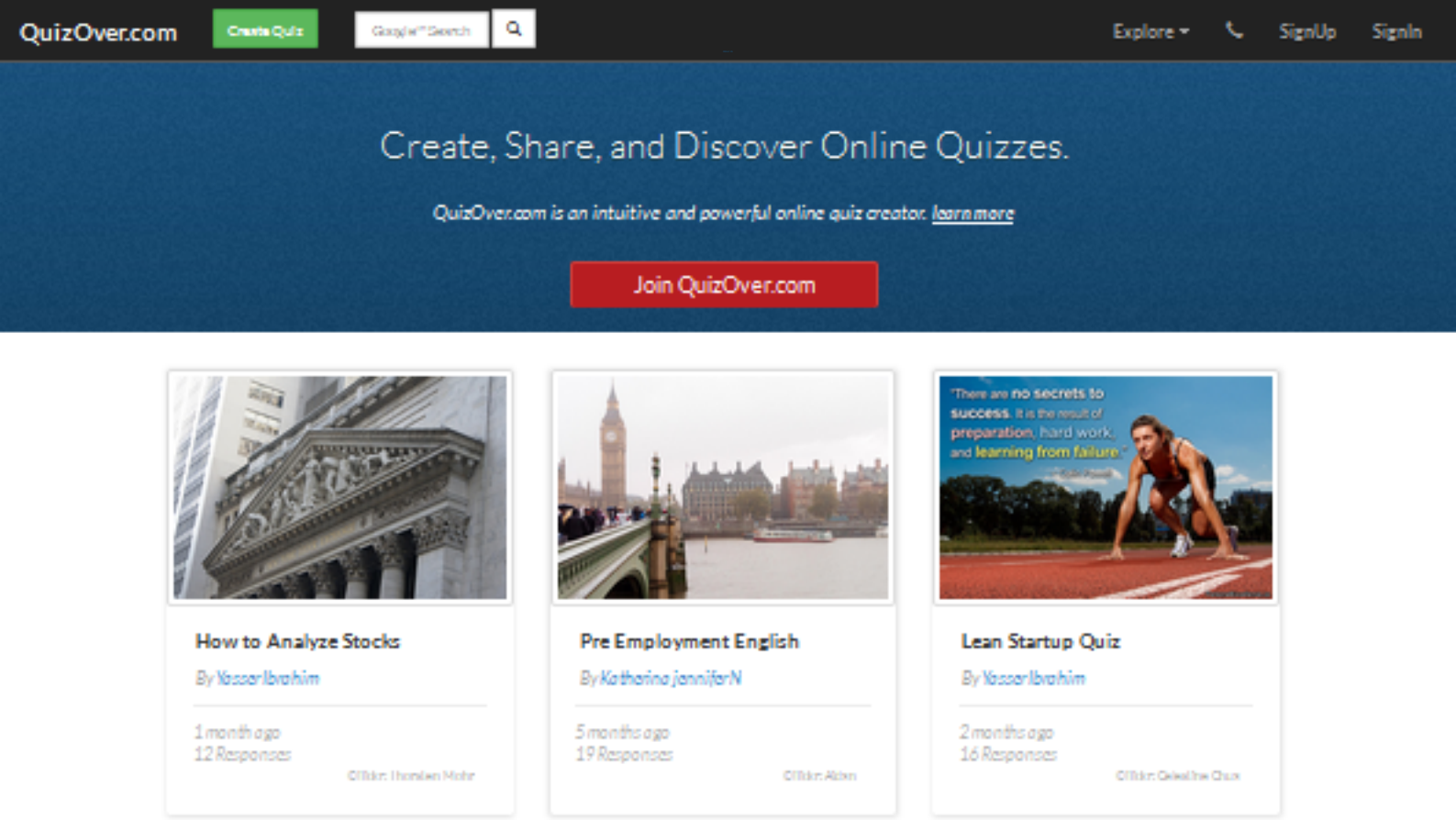
****

Enterprise Frameworks Java Persistence API

JPA

Author: JavaChamp Team

Senior Java Developer @QuizOver.com



Powered by QuizOver.com

The Leading Online Quiz & Exam Creator

Create, Share and Discover Quizzes & Exams

http://www.quizover.com

Disclaimer

All services and content of QuizOver.com are provided under QuizOver.com terms of use on an "as is" basis, without warranty of any kind, either expressed or implied, including, without limitation, warranties that the provided services and content are free of defects, merchantable, fit for a particular purpose or non-infringing.

The entire risk as to the quality and performance of the provided services and content is with you.

In no event shall QuizOver.com be liable for any damages whatsoever arising out of or in connection with the use or performance of the services.

Should any provided services and content prove defective in any respect, you (not the initial developer, author or any other contributor) assume the cost of any necessary servicing, repair or correction.

This disclaimer of warranty constitutes an essential part of these "terms of use".

No use of any services and content of QuizOver.com is authorized hereunder except under this disclaimer.

The detailed and up to date "terms of use" of QuizOver.com can be found under:

http://www.QuizOver.com/public/termsOfUse.xhtml

eBook Content License

Table of Contents

Quiz Permalink: http://www.quizover.com/question/jpa

Author Profile: http://www.quizover.com/user/profile/Java.Champ

4. Chapter: Entity Classes

1. Entity Classes Questions

4.1.1. You want to write a JPA Entity class to model a databank table name...

Author: Yasser Ibrahim

You want to write a JPA Entity class to model a databank table named COOPERATE_USER. Each user is uniquely identified in this table by his or her social security number SSN. Along with SSN the system keeps user name, job, address and birth date.

How to declare such JPA Entity class?

Please choose all the answers that apply:

- Write a CooperateUser public class
- Annotate the CooperateUser class with @Table(name=" COOPERATE_USER")
- Annotate the CooperateUser class with @Entity
- The CooperateUser class must implement Serializable
- Define private attributes ssn, job, name, addess and birthdate
- Annotate the ' name' attibute with @Id

Check the answer of this question online at QuizOver.com:
Question: how to write ejb entity class?

Flashcards:
http://www.quizover.com/flashcards/how-to-write-ejb-entity-class?pdf=1505

Interactive Question:
http://www.quizover.com/question/how-to-write-ejb-entity-class?pdf=1505

4.1.2. How to declare JPA Entity class?

Author: Yasser Ibrahim

How to declare JPA Entity class?

Please choose all the answers that apply:

- Write a private final Entity class named after the mapped table
- Annotate the class with @Entity
- Annotate the primary key attribute or its getter method with @Id
- Define private attributes to map the table columns
- Write the finalize method

Check the answer of this question online at QuizOver.com:
Question: jpa entity example

Flashcards:
http://www.quizover.com/flashcards/jpa-entity-example?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-entity-example?pdf=1505

4.1.3. What is the impact of marking an attribute in a JPA Entity class wi...

Author: Yasser Ibrahim

What is the impact of marking an attribute in a JPA Entity class with @Transient annotation?

Please choose only one answer:

- This attribute will should not be serialized by the EJB container
- This attribute will should not be persisted by the EJB container
- This attribute will should not be garbage collected by the EJB container

Check the answer of this question online at QuizOver.com:
Question: @entity javadoc

Flashcards:
http://www.quizover.com/flashcards/entity-javadoc?pdf=1505

Interactive Question:
http://www.quizover.com/question/entity-javadoc?pdf=1505

4.1.4. Given the code snippet below, which of the following statements are...

Author: Yasser Ibrahim

Given the code snippet below, which of the following statements are true?

```
@PersistenceContext
 private EntityManager entityManager;

 public void registerUser() {
  User usr = new User();   // Line 1
  usr.setName("JavaChamp"); // Line 2
  entityManager.persist(usr); // Line 3
 }
```

Please choose all the answers that apply:

- At line 1 the user instance is a managed Entity
- At line 1 the user instance is a new Entity
- At line 1 a new record will be inserted in the user table
- At line 2 the entityManager will immediately reflect user name change to the database
- At line 3 the entityManager will immediately reflect user name change to the database

Check the answer of this question online at QuizOver.com:
Question: container managed entity bean

Flashcards:
http://www.quizover.com/flashcards/container-managed-entity-bean?pdf=1505

Interactive Question:
http://www.quizover.com/question/container-managed-entity-bean?pdf=1505

4.1.5. Given the code snippet below, which of the following statements are...

Author: Yasser Ibrahim

Given the code snippet below, which of the following statements are true?

```
@PersistenceContext
 private EntityManager entityManager;

 public void updateUserProfile(Long usrId) {
  User usr = entityManager.find(User.class,usrId); // Line 1
  usr.setName("JavaChamp"); // Line 2
  entityManager.merge(usr); // Line 3
  entityManager.persist(usr); // Line 4

 }
```

Please choose all the answers that apply:

- At line 1 the user instance is a managed Entity
- At line 1 the user instance is a detached Entity
- At line 1 the find method will return null if the input userID is not found
- line 3 is mandatory to synchronize the changes done at line 2 to the database
- line 4 is mandatory to synchronize the changes done at line 2 to the database

Check the answer of this question online at QuizOver.com:
Question: detached entity passed to persist manytomany

Flashcards:
http://www.quizover.com/flashcards/detached-entity-passed-to-persist-manytomany?pdf=1505

Interactive Question:
http://www.quizover.com/question/detached-entity-passed-to-persist-manytomany?pdf=1505

4.1.6. In which life cycle state the JPA entity class is synchronized with...

Author: Yasser Ibrahim

In which life cycle state the JPA entity class is synchronized with the database?

Please choose only one answer:

- New
- Managed
- Synchronized
- Detached
- Removed

Check the answer of this question online at QuizOver.com:
Question: jpa entity synchronized life cycle states

Flashcards:
http://www.quizover.com/flashcards/jpa-entity-synchronized-life-cycle-states?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-entity-synchronized-life-cycle-states?pdf=1505

4.1.7. What causes the entity manager to move an Entity instance from the ...

Author: Yasser Ibrahim

What causes the entity manager to move an Entity instance from the detached to the managed state?

Please choose only one answer:

- Invoke the persist() method of the entity manager on the instance
- Invoke the merge() method of the entity manager on the instance
- Invoke the manage() method of the entity manager on the instance
- Invoke the remove() method of the entity manager on the instance

Check the answer of this question online at QuizOver.com:
Question: jpa entity manager merge

Flashcards:
http://www.quizover.com/flashcards/jpa-entity-manager-merge?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-entity-manager-merge?pdf=1505

4.1.8. What causes the entity manager to move an Entity instance from the ...

Author: Yasser Ibrahim

What causes the entity manager to move an Entity instance from the managed to the detached state?

Please choose only one answer:

- Invoke the detach() method of the entity manager on the instance
- Invoke the remove() method of the entity manager on the instance
- Invoke the unsynchronize() method of the entity manager on the instance
- when the associated persistence context ends at the end of the transaction boundary.

Check the answer of this question online at QuizOver.com:
Question: jpa entity manager detached state

Flashcards:
http://www.quizover.com/flashcards/jpa-entity-manager-detached-state?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-entity-manager-detached-state?pdf=1505

4.1.9. What are the possible state transitions for an Entity instance ?

Author: Yasser Ibrahim

What are the possible state transitions for an Entity instance ?

Please choose all the answers that apply:

- from managed to detached
- from detached to removed
- from detached to new
- from new to detached
- from managed to removed

Check the answer of this question online at QuizOver.com:
Question: jpa state transitions

Flashcards:
http://www.quizover.com/flashcards/jpa-state-transitions?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-state-transitions?pdf=1505

4.1.10. which of the following is NOT a jpa entity manager method?

Author: Yasser Ibrahim

which of the following is NOT a jpa entity manager method?

Please choose only one answer:

- persist
- flush
- contains
- detach
- find

Check the answer of this question online at QuizOver.com:
Question: jpa entity manager methods

Flashcards:
http://www.quizover.com/flashcards/jpa-entity-manager-methods?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-entity-manager-methods?pdf=1505

4.1.11. Which method of the jpa entity manager would you use to force synch...

Author: Yasser Ibrahim

Which method of the jpa entity manager would you use to force synchronizing the persistence context from the database?

Please choose only one answer:

- synchronize()
- flush()
- refresh()
- persist()

Check the answer of this question online at QuizOver.com:
Question: jpa entity manager refresh

Flashcards:
http://www.quizover.com/flashcards/jpa-entity-manager-refresh?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-entity-manager-refresh?pdf=1505

4.1.12. Which method of the jpa entity manager would you use to force synch...

Author: Yasser Ibrahim

Which method of the jpa entity manager would you use to force synchronizing the database with the entities of the persistence context?

Please choose only one answer:

- synchronize()
- flush()
- refresh()
- persist()

Check the answer of this question online at QuizOver.com:
Question: jpa entity manager flush

Flashcards:
http://www.quizover.com/flashcards/jpa-entity-manager-flush?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-entity-manager-flush?pdf=1505

4.1.13. What is considered optional while you define a persistence unit?

Author: Yasser Ibrahim

What is considered optional while you define a persistence unit?

Please choose all the answers that apply:

- persistence.xml configuration file
- Entity classes annotated with @Entity
- The ORM definition (object relational mapping)
- entity manager configuration (e.g. toplink, hibernate)

Check the answer of this question online at QuizOver.com:
Question: persistence unit definition

Flashcards:
http://www.quizover.com/flashcards/persistence-unit-definition?pdf=1505

Interactive Question:
http://www.quizover.com/question/persistence-unit-definition?pdf=1505

4. Chapter: Entity association

1. Entity association Questions

4.1.1. Which of the following statements about the fetch modes in Entity b...

Author: JavaChamp Team

Which of the following statements about the fetch modes in Entity beans is FALSE?

Please choose only one answer:

- There are two fetch modes : EAGER and LAZY
- The default fetching mode for a field in an Entity bean annotated by @Basic is LAZY
- The default fetching mode for a field in an Entity bean annotated by @OneToMany is LAZY
- @Lob annotation does not have a default fetch mode

Check the answer of this question online at QuizOver.com:
Question: jpa eager loading

Flashcards:
http://www.quizover.com/flashcards/jpa-eager-loading?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-eager-loading?pdf=1505

4.1.2. What is TRUE about the cascading and cascade mode attributes in Ent...

Author: JavaChamp Team

What is TRUE about the cascading and cascade mode attributes in Entity Beans?

Please choose all the answers that apply:

- Cascade mode attributes can be specified for the association annotaions (like @OneToMany) in an entity bean
- The cascading direction is from the target entity to the source entity
- PERSIST, DELETE and REFRESH are cascading mode attributes
- Refresh cascade causes to refresh the target entities of a relationship when refresh is invoked on the source entity of the relationship

Check the answer of this question online at QuizOver.com:
Question: jpa persist, merge, remove, refresh and all cascade modes

Flashcards:
http://www.quizover.com/flashcards/jpa-persist-merge-remove-refresh-and-all-cascade-modes?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-persist-merge-remove-refresh-and-all-cascade-modes?pdf=1505

4.1.3. what are the JPA @Entity association attributes?

Author: Yasser Ibrahim

what are the JPA @Entity association attributes?

Please choose all the answers that apply:

- Association validation
- Association multiplicity
- Association cascade behavior
- Association direction

Check the answer of this question online at QuizOver.com:
Question: jpa association attributes

Flashcards:
http://www.quizover.com/flashcards/jpa-association-attributes?pdf=1505

Interactive Question:
http://www.quizover.com/question/jpa-association-attributes?pdf=1505

4.1.4. Which @Entity association cardinality and direction is implemented ...

Author: Yasser Ibrahim

Which @Entity association cardinality and direction is implemented in the following code snippet?

```
@Entity
public class Yard {
 @Id
 private int yardNo;
}

@Entity
public class House {
 @Id
 private int houseNo;
 @OneToOne
 private Yard backYard;
}
```

Please choose only one answer:

- @OneToOne unidirectional
- @OneToOne bidirectional
- @OneToOne no directional attribute specified

Check the answer of this question online at QuizOver.com:
Question: onetoone unidirectional association

Flashcards:
http://www.quizover.com/flashcards/onetoone-unidirectional-association?pdf=1505

Interactive Question:
http://www.quizover.com/question/onetoone-unidirectional-association?pdf=1505

4.1.5. Which @Entity association cardinality and direction is implemented ...

Author: Yasser Ibrahim

Which @Entity association cardinality and direction is implemented in the following code snippet?

```
@Entity
public class Yard {
 @Id
 private int yardNo;
 @OneToOne(mappedBy="backYard")
 private House aHouse;
}

@Entity
public class House {
 @Id
 private int houseNo;
 @OneToOne
 private Yard backYard;
}
```

Please choose only one answer:

- @OneToOne unidirectional
- @OneToOne bidirectional
- @OneToOne no directional attribute specified

Check the answer of this question online at QuizOver.com:
Question: association onetoone bidirectional

Flashcards:
http://www.quizover.com/flashcards/association-onetoone-bidirectional?pdf=1505

Interactive Question:
http://www.quizover.com/question/association-onetoone-bidirectional?pdf=1505

4.1.6. What's true about the following @Entity association between House a...

Author: Yasser Ibrahim

What's true about the following @Entity association between House and Yard?

```
@Entity
public class Yard {
 @Id
 private int yardNo;
 @OneToOne(mappedBy="backYard")
 private House aHouse;
}

@Entity
public class House {
 @Id
 private int houseNo;
 @OneToOne
 private Yard backYard;
}
```

Please choose all the answers that apply:

- It's OneToOne unidirectional association
- It's OneToOne bidirectional association
- The association owner is the House class
- The association owner is the Yard class

Check the answer of this question online at QuizOver.com:
Question: entity association onetoone bidirectional

Flashcards:
http://www.quizover.com/flashcards/entity-association-onetoone-bidirectional?pdf=1505

Interactive Question:
http://www.quizover.com/question/entity-association-onetoone-bidirectional?pdf=1505

4.1.7. Which @Entity association attributes are implemented in the followi...

Author: Yasser Ibrahim

Which @Entity association attributes are implemented in the following code snippet?

```
@Entity
public class Window {
 @Id
 private int winNo;
 @ManyToOne
 private House aHouse;
}

@Entity
public class House {
 @Id
 private int houseNo;
 @OneToMany(mappedBy="aHouse")
 private List<Window> windows;
}
```

Please choose only one answer:

- @OneToMany unidirectional
- @OneToMany bidirectional
- @OneToMany no directional attribute specified

Check the answer of this question online at QuizOver.com:
Question: entity association onetomany bidirectional

Flashcards:
http://www.quizover.com/flashcards/entity-association-onetomany-bidirectional?pdf=1505

Interactive Question:
http://www.quizover.com/question/entity-association-onetomany-bidirectional?pdf=1505

4.1.8. What's true about the following @Entity association between House a...

Author: Yasser Ibrahim

What's true about the following @Entity association between House and Window?

```
@Entity
public class Window {
 @Id
 private int winNo;
 @ManyToOne
 private House aHouse;
}

@Entity
public class House {
 @Id
 private int houseNo;
 @OneToMany(mappedBy="aHouse")
 private List<Window> windows;
}
```

Please choose all the answers that apply:

- It's OneToMany unidirectional association
- It's OneToMany bidirectional association
- The association owner is the House class
- The association owner is the Window class

Check the answer of this question online at QuizOver.com:
Question: entity association onetomany bidirectional example

Flashcards:
http://www.quizover.com/flashcards/entity-association-onetomany-bidirectional-example?pdf=1505

Interactive Question:
http://www.quizover.com/question/entity-association-onetomany-bidirectional-example?pdf=1505

4.1.9. Which @Entity association attributes are implemented in the followi...

Author: Yasser Ibrahim

Which @Entity association attributes are implemented in the following code snippet?

```
@Entity
public class Course {
 @Id
 private int courseNo;
 @ManyToMany
 private List<Student> studentList;
}

@Entity
public class Student {
 @Id
 private int StudentNo;
 @ManyToMany(mappedBy="studentList")
 private List<Course> CourseList;
}
```

Please choose only one answer:

- @ManyToMany unidirectional lazy fetch mode
- @ManyToMany bidirectional lazy fetch mode
- @ManyToMany bidirectional eager fetch mode
- @ManyToMany unidirectional eager fetch mode

Check the answer of this question online at QuizOver.com:
Question: entity association manytomany bidirectional

Flashcards:
http://www.quizover.com/flashcards/entity-association-manytomany-bidirectional?pdf=1505

Interactive Question:
http://www.quizover.com/question/entity-association-manytomany-bidirectional?pdf=1505

Question No :1
Which methods can't be overridden in the JSP page ?

(Choose correct one from multiple below)
1. jspDestroy()
2. jspInit()
3. _jspService()
4. getParameter()
correct is :3

Question No :2
The tag %@include file=?.....? % > helps in including ?

(Choose correct one from multiple below)
1. readOnly
2. dynamic
3. static
4. None of the above
correct is :3

Question No :3
The jsp:useBean attribute is used to indicate the Serialized bean?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :4
Is data in request object avilable after response.sendRedirect();
(Choose correct one from multiple below)
1. yes avilable
2. Not avilable
3. Can't say
4. None of the above
correct is :2

Question No :5
The response.sendRedirect("??..?); is the response implicit object to redirect the browser to the different resource?

(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :6
The jsp:plugin tag is used to insert the browser-specific OBJECTS and EMBED elements?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :7
Which tag in the jsp is used to define the error page?
(Choose correct one from multiple below)
1. <%@page isErrorPage="yes" %>
2. <%@page isErrorPage="true" %>
3. <%@page isErrorPage="false" %>
4. None of the above
correct is :2

Question No :8
Can we make use of a ServletOutputStream object from the JSP page?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. None of the above
correct is :1

Question No :9
Adding a meta, usage like <meta name=?das? contents=?no index?> tag to the jsp prevents it from being indexed by the search engines like Yahoo and Google

(Choose correct one from multiple below)
1. true
2. false
3. can't say

4. none of the above
correct is :1

Question No :10
To show money format on the JSP page we use _____ class ?

(Choose correct one from multiple below)
1. NumberFormat
2. StringBuffer
3. CurrencyFormat
4. AirthmaticFormat
correct is :1

Question No :11
The error message displayed when the page is not found at the correct location?
(Choose correct one from multiple below)
1. 500
2. 404
3. 505
4. 440
correct is :2

Question No :12
Request.getServerName () is used to get the name of the server on which the Jsp is running?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :13
Which of the following statements is true about JSP tag library?

(Choose correct one from multiple below)
1. It defines the standard tag that works the same everywhere
2. It is a single library and we can use it in multiple jsp containers

3. It has support for the common structural tasks like iteration and condition.
4. All of the above
correct is :4

Question No :14
Which of the scripting of JSP not putting content into service method of the converted servlet ?

(Choose correct one from multiple below)
1. Declarations
2. Scriptlets
3. expressions
4. None of the above
correct is :1

Question No :15
Can we implement an interface in JSP ?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :2

Question No :16
ServletContext class gives information about the _____?
(Choose correct one from multiple below)
1. request
2. session
3. container
4. none of the above
correct is :3

Question No :17
PageContext class gives information about the _____?
(Choose correct one from multiple below)
1. application
2. session
3. request
4. response
correct is :3

Question No :18
While using request.getRequestDispatcher(path) we need to give the _____ path of the object ?
(Choose correct one from multiple below)
1. absolute
2. relative
3. can't say
4. none of the above
correct is :2

Question No :19
While using context.getRequestDispatcher(path) we need to give the _____ path of the object ?
(Choose correct one from multiple below)
1. absolute
2. relative
3. can't say
4. none of the above
correct is :1

Question No :20
URL instance represents the location of a resource?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :21
URLConnection instance represents a link for accessing or communicating with the resource at the location?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :22
When u try to redirect a page after you already have written something in your page, the error encountered is
?Response has already been committed error? or popularly called 402 error.?
(Choose correct one from multiple below)

1. true
2. false
3. none of the above
4. none of the above
correct is :1

Question No :23
The wrapper function like
<%!
String blanknull(String s) {
return (s == null) ? "" : s;
}
%>
then use it inside your JSP form, like
<input type="text" name="size"
value="<%=blanknull(size)% >" >
prevents the word_____ from apperaring in an HTML page.

(Choose correct one from multiple below)
1. Null
2. Not Null
3. space
4. None of the above
correct is :1

Question No :24
The code block
<%@ page isErrorPage="true" %>
<%
out.println(" ");
PrintWriter pw = response.getWriter();
exception.printStackTrace(pw);
out.println(" ");
%>
Helps in printing the _____ of exception on JSP.
(Choose correct one from multiple below)
1. stacktrace
2. message
3. error message
4. none of the above
correct is :1

Question No :25
The code block
<%@ page isErrorPage="true" %> is used for ?

(Choose correct one from multiple below)
1. print error
2. print output
3. get input
4. none of the above
correct is :1

Question No :26
Which of the following correctly defines JSP technology ?

(Choose correct one from multiple below)
1. JSP page is a text-based document that describes how to process a request to create a response.
2. JSP page is a text-based document that describes how to process a to response create a request.
3. JSP page is a xml-based document that describes how to process a request to create a response.
4. JSP page is a xml-based document that describes how to process a to response create a request.
correct is :1

Question No :27
URL encoding is the method of replacing all the spaces and other extra characters into their corresponding _____ Characters ?

(Choose correct one from multiple below)
1. Hex
2. Binary
3. Octal
4. Decimal
correct is :1

Question No :28
All the _____ data is kept at the application server _____ data is kept at the web server ?

(Choose correct one from multiple below)
1. dynamic, static
2. static, dynamic
3. HTML, Servlet
4. Servlet, HTML
correct is :1

Question No :29
Under JSP 1.0 , the implicit object which can be used for reference is ?this?. It is used to refer to the _____ ?

(Choose correct one from multiple below)
1. Servlet generated by the JSP page
2. Calling the previous JSP page
3. Servlet called by the jsp page
4. None of the above
correct is :1

Question No :30
Which of the following are the implicit objects in JSP
1. Application, out
2. config, exception
3. page, pageContext
4. request, response, session

(Choose correct one from multiple below)
1. 1, 2
2. 2, 3
3. 2, 3, 4
4. 1, 2, 3, 4
correct is :4

Question No :31
Can a JSP process HTML form data ?
(Choose correct one from multiple below)
1. true

2. false
3. can't say
4. none of the above
correct is :1

Question No :32
State true or false -- It is possible to call an external application like MSWord by the click on the JSP page or Servlet?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :33
What tools can be used to generate the multiple views in jsp ?

(Choose correct one from multiple below)
1. xalan-J
2. Saxon
3. Jdk1.5
4. None of the above
correct is :1

Question No :34
What is the possible way of communication between applet and Servlet ?

(Choose correct one from multiple below)
1. HTTP Communication (Text-based and object-based)
2. Socket Communication
3. RMI Communication
4. All of the above
correct is :4

Question No :35
JSP technology is extensible?
(Choose correct one from multiple below)

1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :36
What is a translation unit ?

(Choose correct one from multiple below)
1. JSP page can include the contents of other HTML pages or other JSP files.
2. When the JSP engine is presented with such a JSP page it is converted to one Servlet class and this is called a translation unit,
3. In a translation unit is that page directives affect the whole unit, one variable declaration cannot occur in the same unit more than once, the standard action jsp:useBean cannot declare the same bean twice in one unit.
4. All of the above
correct is :2

Question No :37
HttpSession.setMaxInactiveInterval() can manage the inactivity lease period on a per-session basis?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :38
Some Operating Systems have a limitation on the Max Length of the URL, which are normally _____ characters?
(Choose correct one from multiple below)
1. 566
2. 260
3. 256
4. 255
correct is :4

Question No :39
Fill in the blanks -- Sharing of session data across multiple load balanaced web servers is called

_____ ?
(Choose correct one from multiple below)
1. Clustering
2. ShareWeb
3. loadBalancer
4. None of the above
correct is :1

Question No :40
The session tracking in the JSP can be done by
1. URL rewriting
2. Cookies
3. User-Authorization
4. Hidden value

(Choose correct one from multiple below)
1. 1,3,4 only
2. 2,3,4 only
3. 1,2,4 only
4. 1,2,3,4
correct is :3

Question No :41
From the current Browser window, if we open a new Window, then it referred to as Multiple Windows. Are Sessions properties are maintained across all these windows, even though they are operating in multiple windows ?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :42
If the session object is invalidated can it be again regained or refreshed?
(Choose correct one from multiple below)
1. Yes
2. No
3. Can't say
4. none of the above

correct is :2

Question No :43

What is the role played by JSP in the MVC architecture ?

(Choose correct one from multiple below)

1. Model
2. View
3. Controller
4. None of the above

correct is :2

Question No :44

The listener is notified when the Servlet context (i.e., the Web application) is initialized and destroyed ?

(Choose correct one from multiple below)

1. Servlet context attribute
2. Servlet context listeners
3. Session Attribute
4. Session Listeners

correct is :2

Question No :45

The_____ listener is notified when attributes are added to, removed from, or replaced in the Servlet context?

(Choose correct one from multiple below)

1. Session context Listeners
2. Servlet context attribute
3. Session Listeners
4. Session Attribute

correct is :2

Question No :46

The _____ listeners are notified when session objects are created, invalidated, or timed out.

(Choose correct one from multiple below)

1. Servlet context listeners

2. Session Listeners
3. Servlet context attribute
4. Session Attribute
correct is :2

Question No :47
Arrange the lifecycle phases of JSP in correct order1. Page translation: -page is parsed, and a java file which is a Servlet is created. 2. Page compilation: page is compiled into a class file 3. Page loading : This class file is loaded. 4. Create an instance :- Instance of Servlet is created 5. jspInit() method is called 6. _jspService is called to handle service calls 7. _jspDestroy is called to destroy it when the Servlet is not required.

(Choose correct one from multiple below)
1. 4,5,3,6,2,7,1
2. 1,2,3,4,5,6,7
3. 1,4,3,2,5,6,7
4. 3,2,1,4,5,6,7
correct is :2

Question No :48
The JspPage interface defines the _____ and ____ method which the page writer can use in their pages and are invoked in much the same manner as the ____ and _____ methods of a servlet.

(Choose correct one from multiple below)
1. jspInit(), jspDestroy(), init() , destroy()
2. init(),jspDestroy(),jspInit(),destroy()
3. destroy(),jspDestroy(),jspInit(),init()
4. init(),destroy(),jspInit(), jspDestroy()
correct is :1

Question No :49
Which of the following statements is true regarding the scope of ?request? in JSP ?

(Choose correct one from multiple below)

1. Objects with request scope are accessible from pages processing the same request where they were created.
2. All references to the object shall be released after the request is processed; in particular, if the request is forwarded to a resource in the same runtime, the object is still reachable.
3. References to objects with request scope are stored in the request object.
4. All of the above.
correct is :1

Question No :50
There is a limit to the max amount of data that can be stored in the session object ?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :2

Question No :51
The maximum length of the session Id(length identifier) is _____?
(Choose correct one from multiple below)
1. 4k
2. 2k
3. 6k
4. 8k
correct is :1

Question No :52
The browsers are only required to accept_____ cookies per site?
(Choose correct one from multiple below)
1. 20
2. 30
3. 22
4. 24
correct is :1

Question No :53
The maximum age of the cookie in JSP can be set by ?

(Choose correct one from multiple below)
1. cookie.setAgeMax (int seconds)
2. cookie.setAgeMax (float seconds)

3. cookie.setMaxAge(float seconds)
4. cookie.setMaxAge(int seconds)
correct is :4

Question No :54
The ___ indicates that the cookie will expire once the browser is closed?
(Choose correct one from multiple below)
1. cookie.setMaxAge(-1)
2. cookie.setMaxAge(0)
3. cookie.setMaxAge(365)
4. None of the above
correct is :1

Question No :55
The ____ is used to delete a cookie?
(Choose correct one from multiple below)
1. cookie.setMaxAge(0)
2. cookie.setMaxAge(-1)
3. cookie.setMaxAge(-2)
4. None of the above
correct is :1

Question No :56
Which of the following correctly describes the sequence of calling the methods for preventing browser from caching the details. i.response.setHeader("Cache-Control","no-store");
ii. response.setHeader("Pragma","no-cache"); iii.response.setDateHeader ("Expires", 0);
(Choose correct one from multiple below)
1. i,ii,iii
2. ii,i,iii
3. iii,ii,i
4. None of the above
correct is :1

Question No :57
The < % return; % > simply aborts the processing of JSP?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :58

The _____ is used to call other Servlet or JSP from the current Servlet ?

(Choose correct one from multiple below)

1. RequestDispatcher
2. ResponseDispatcher
3. ServletInvoker
4. None of the above

correct is :1

Core Spring 3.0 Certification Mock Exam

Question

Container

Question 1

Given the following Spring configuration file, what is the correct answer:

```
<bean class="com.spring.service.MyServiceImpl">
	<property name="repository" ref="jpaDao"/>
</bean>

<bean class="com.spring.repository.JpaDao"/>
```

1. The first declared bean MyServiceImpl is missing an id must be named myService
2. The second declared bean JpaDao is missing an id must be named jpaDao
3. Answers 1 and 2 are both rights
4. Answers 1 and 2 are both wrong

Question 2

Given the Spring configuration file, which are the correct statements?

```
<bean class="com.spring.service.BankServiceImpl"
                    p:bankName="NationalBank">
</bean>
```

1. The p namespace has to be declared
2. Bean id is bankServiceImpl
3. The BankServiceImpl references a NationalBank bean
4. NationalBank is a scalar value

Question 3

How is named the bean that is defined in the following configuration class. Select a single answer.

```
@Configuration
public class ApplicationConfig {

	@Autowired
	private DataSource dataSource;

	@Bean
	ClientRepository clientRepository() {
		ClientRepository accountRepository = new JpaClientRepository();
		accountRepository.setDataSource(dataSource);
		return accountRepository;
```

```
        }
}
```

1. JpaClientRepository
2. jpaClientRepository
3. clientRepository
4. `Two beans are defined : a data souce and a repository`

Question 4

How could you externalize constants from a Spring configuration file or a Spring annotation into a .properties file? Select one or more answers

1. By using the <util:constant /> tag
2. By declaring the ConstantPlaceholderConfigurer bean post processor
3. By using the <context:property-placeholder /> tag
4. By using the c: namespace

Question 5

What statement is not correct in live environment? Select a unique answer.

1. Constuctor and properties autowiring in the same bean are not compatible
2. A bean should have a default or a no-args constructor
3. The <constructor-arg> tag could take type, name and index to reduce ambiguity
4. None of the above
5. All of the above

Question 6

What are the right affirmations about the @PostConstruct, @Resource and the @PreDestroy annotations?

1. Those annotations are specified in the JSR-250
2. The Spring Framework embedded those annotation
3. The <context:component-scan> tag enable them
4. The <context:annotation-config > tag enable them
5. Declaring the CommonAnnotationBeanPostProcessor enable them

Question 7

What is/are typically case(s) where you usually need to manually instanciated an ApplicationContext?

1. In a web application
2. In an integration test running with the SpringJUnit4ClassRunner
3. In a standalone application started with a main method
4. None of the above

Question 8

Select the right statement about referring a Spring configuration file inside the package com.example.myapp in the below example?

```
ApplicationContext context = new
ClassPathXmlApplicationContext("classpath:/com.example.myapp.config.xml");
```

1. The classpath: prefix could be omit
2. Package name with dot is not well formatted using the dot character
3. The slash character preceding com.example could be omit
4. All of the above
5. None of the above

Question 9

How to auto-inject into a field a bean by its name? Select one or more response.

1. With the name attribute of the @Autowired annotation
2. By using the single @Qualifier annotation
3. By using both the @Autowired and the @Qualifier spring annotations
4. By using the @Autowired annotation and naming the field with the bean name

Question 10

What are the main advantages of using interfaces when designing business services? Select one or more answers.

1. Mocking or stubbing the service
2. Be able to use the Spring auto-injection
3. Can do dependency checking
4. Loosely coupled code

Question 11

Select one or many correct answers about spring bean life cycle.

1. The method annoted with @PostConstruct is called after bean instantiation and before properties setting of the bean
2. The method @PreDestroy of a prototype bean is called when the bean is garbage collected
3. The init() method declared in the init-method attribute of a bean is called before the afterPropertiesSet callback method of the InitializingBean interface
4. The method annotated with @PostConstruct is called before before the afterPropertiesSet callback method of the InitializingBean interface

Question 12

Given the following configuration class, what are correct affirmations? Select one or more answers.

```java
public class ApplicationConfig {

	private DataSource dataSource;

	@Autowired
	public ApplicationConfig(DataSource dataSource) {
		this.dataSource = dataSource;
	}

	@Bean(name="clientRepository")
	ClientRepository jpaClientRepository() {
		return new JpaClientRepository();
	}
}
```

1. @Configuration annotation is missing
2. Default or no-arg constructor is missing
3. @Bean name is ambiguous
4. @Bean scope is prototype

Question 13

What are the features of the XML <context: namespace? Select one or many answers.

1. @Transactional annotation scanning
2. @Aspect annotation detection enabling
3. @Autowired annotation enabling
4. @Component annotation scanning

Test

Question 14

Select one or more correct statements about developing integration test with Spring support.

1. A new Spring context is created for each test class
2. To get a reference on the bean you want to test, you have to call the getBean() method of the Spring context
3. Spring context configuration could be inherits from the super class
4. The Spring context configuration file has to be provided to the @ContextConfiguration annotation

Question 15

What are the main advantage(s) for using Spring when writing integration tests?

1. Reuse Spring configuration files of the application
2. Create mock or stub
3. Be able to use the rollback after the test pattern
4. Use dependency injection

Question 16

What are the main advantage(s) for using Spring when writing unit tests?

1. Reuse Spring configuration files of the application
2. Use dependency injection
3. Provide some mocks for servlet classes
4. All of the above
5. None of the above

Question 17

What is right about the spring test module?

1. It provides an abstraction layer for the main open source mock frameworks
2. Provides the @Mock annotation
3. It dynamically generates mock objects
4. All of the above
5. None of the above

Question 18

Select correct statement(s) about transactional support of the spring test module.

1. Transaction manager could be set within the @TransactionConfiguration annotation
2. Method annotated with @Before is executed outside of the test's transaction
3. Spring test may rollback the transaction of a service configured with the REQUIRES_NEW propagation
4. The transaction of a method annotated with the @Rollback annotation with its default values is rolled back after the method has completed

AOP

Question 19

Considering 2 classes AccountServiceImpl and ClientServiceImpl. Any of these 2 classes inherits from each other. What is the result of the pointcut expressions?

```
   execution(* *..AccountServiceImpl.update(..))
&& execution(* *..ClientServiceImpl.update(..))
```

1. Matches pubic update methods of the 2 classes, whatever the arguments
2. Matches any update methods of the 2 classes , whatever the arguments and method visibility
3. Matches any update methods of the 2 classes , with one more arguments and whatever method visibility
4. No joint point is defined

Question 20

Using the Spring AOP framework, what is the visibility of the method matches by the following join point?

```
@Pointcut("execution(* *(..))")
private void anyOperation() {};
```

1. All methods, whereas there visibility
2. All methods, except private method
3. Protected and public methods
4. Public methods

Question 21

What are the 2 correct statements about AOP proxy?

1. AOP proxies are created by Spring in order to implement the aspect contracts
2. AOP proxies are always created with a JDK dynamic proxy
3. Only classes that implements a least one interface could be proxied
4. All methods could be proxied
5. Proxies are created by a BeanPostProcessor

Question 22

What is an after throwing advice? Select a unique answer.

1. Advice that could throw an exception
2. Advice to be executed if a method exits by throwing an exception
3. Advice that executes before a join point
4. Spring does not provide this type of advice

Question 23

What is an after returning advice? Select a unique answer.

1. Advice to be executed regardless of the means by which a join point exits
2. Advice that surrounds a method invocation and can perform custom behavior before and after the method invocation
3. Advice to be executed before method invocation
4. Advice to be executed after a join point completes without throwing an exception

Question 24

What is an advice? Select a unique answer.

1. An action taken by an aspect at a particular join point
2. A point during the execution of a program
3. An aspect and a pointcut
4. A predicate that matches join points

Question 25

What is a pointcut? Select a unique answer.

1. Code to execute at a join point
2. An expression to identify joinpoints
3. An advice and a jointpoint
4. None of the above

Question 26

Select method's signatures that match with the following pointcut:

execution(* com.test.service..*.*(*))

1. void com.test.service.MyServiceImpl#transfert(Money amount)
2. void com.test.service.MyServiceImpl#transfert(Account account, Money amount)
3. void com.test.service.account.MyServiceImpl#transfert(Money amount)
4. void com.test.service.account.MyServiceImpl#transfert(Account account, Money amount)
5. None of the above

Question 27

What are the unique correct answers about Spring AOP support?

1. An advice could proxied a constructor's class
2. A point cut could select methods that have a custom annotation
3. Static initialization code could be targeted by a point cut
4. Combination of pointcuts by &&, || and the ! operators is not supported

Question 28

Using the Spring AOP framework, what are the joinpoint methods of the following pointcut expressions?

execution(public * *(..))

1. The execution of all public method
2. The execution of all public method returning a value
3. The execution of all public method having at least one parameter

4. The execution of all public method in class belonging to the default java package

Data Access

Question 29

Why is it a best practice to mark transaction as read-only when code does not write anything to the database? Select one or more answers.

1. It is mandatory for using Spring exception translation mechanism
2. May be improve performance when using Hibernate
3. Spring optimizes its transaction interceptor
4. Provides safeguards with Oracle and some other databases

Question 30

What data access technology is supported by the Spring framework? Select one or more answers.

1. JDBC
2. NoSQL
3. Hibernate
4. JPA

Question 31

What is not provided by the JdbcTemplate? Select a unique answer.

1. Data source access
2. Open/close data source connection
3. JDBC exception wrapping into DataAccess Exception
4. JDBC statement execution

Question 32

Using JdbcTemplate, what is the Spring provided class you will use for result set parsing and merging rows into a single object? Select a unique answer.

1. RowMapper
2. RowCallbackHandler
3. ResultSetExtractor
4. ResultSetMapper

Question 33

What configuration is supported by the LocalSessionFactoryBean? Select a unique answer.

1. Listing entity classes annoted with @Entity
2. Scanning a package to detect annoted entity classes (with @Entity)
3. Listing hibernate XML mapping configuration file (.hbm.xml)

4. All above

Transaction

Question 34

What is/are incorrect statements about XML declaration of the transaction manager bean? Select one or more answers.

1. The tx namespace provides JTA transaction manager declaration shortcut syntax
2. Id of the bean has to be *transactionManager*
3. Depending the application persistence technology, the HibernateTransactionManager or the DataSourceTransactionManager could be used as bean class
4. Default transaction timeout could be given

Question 35

Assuming @Transactional annotation support is enabled and the transferMoney method is called through a Spring AOP proxy, what is the behavior of the following code sample?

```
@Transactional(propagation=Propagation.REQUIRED)
public void transferMoney(Account src, Account target, double amount) {
	add(src, -amount);
	add(src, amount);
}

@Transactional(propagation=Propagation.REQUIRES_NEW)
public void add(Account account, Double amount) {
	// IMPLEMENTATION
}
```

1. The add() method executes code in a new transaction
2. The add() method uses the transaction of the transferMoney() method
3. When calling the add() method, an exception is thrown
4. Other behavior

Question 36

Does Spring provides programmatic transaction management? Select a unique answer.

1. Yes with the TransactionTemplate class
2. Yes with the TransactionService class
3. Yes using the @Transactional bean post processor
4. No

Question 37

What is the transaction behavior of the PROPAGATION_REQUIRES_NEW mode? Select a unique answer.

1. If a transaction exists, the current method should run within this transaction. Otherwise, it should start a new transaction and run within its own transaction.
2. If a transaction is in progress, the current method should run within the nested transaction of the existing transaction. Otherwise, a new transaction has to be started and run within its own transaction.
3. The current method must start a new transaction and run within its own transaction. If there is an existing transaction in progress, it is suspended.
4. None of the above

Question 38

What is the default rollback policy in transaction management?

1. Rollback for any Exception
2. Rollback for RuntimeException
3. Rollback for checked exceptions
4. Always commit

Sping @MVC

Question 39

What could not return a Spring MVC controller? Select a single answer.

1. An absolute path to the view
2. A logical view name
3. A new JstlView
4. void
5. null value

Question 40

Where do you cannot declare Spring MVC controller? Select one or more answers.

1. In a Spring application context XML configuration file
2. Into the web.xml file of the web application
3. Into the java code by using annotations
4. Into the JSP pages

Question 41

What is the easiest method to write a unit test?

1.
```
void displayAccount(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException
```

2.
```
void displayAccount(HttpServletRequest req, HttpSession Session)
              throws ServletException, IOException
```

3.
```
@RequestMapping("/displayAccount")
String displayAccount(@RequestParam("accountId") int id, Model model)
```

4.
```
@RequestMapping("/displayAccount")
String displayAccount(@PathVariable("accountId") int id, Model model)
```

Spring Security

Question 42

How could you secure MVC controller with Spring Security? Select a unique answer.

1. With the @Secured annotation
2. With the @RolesAllowed annotation
3. In a XML security configuration file
4. All of the above
5. None of the above

Question 43

What are the possible mechanisms provided by Spring Security to store user details? Select one or more correct answers.

1. Database
2. JAAS
3. LDAP
4. Properties file

Question 44

What is true about Spring security configuration and the security namespace? Select one or more correct answers.

1. The access attribute of the intercept-url tag support both EL and constants together.

2. The patterns declared into the intercept-url tag are analyzed from up to bottom. Winning is the first that matches.
3. The patterns declared into the intercept-url tag use by default the java regex syntax.
4. Security rules may applied depending request parameter

Remoting

Question 45

What do you have to do even if you are using the RMI Spring Remoting support? Select one or more correct answers.

1. Implements the Remote interface
2. Extends the RemoteObject class
3. Catching the RemoteException exception
4. Implements the Serializable interface

Question 46

What is exact about the HttpInvokerServiceExporter? Select one or more correct answers.

1. Has to run into a HTPP server as Jetty
2. Could process both POST and GET requests
3. Could be used with any http client as Jakarta Commons HttpClient
4. Could consume SOAP http request

JMS

Question 47

What is the method that is not provided by the JmsTemplate Spring class?

1. convertAndSend
2. onMessage
3. receiveAndConvert
4. setDefaultDestination

Question 48

How could you implement a JMS Listener using the Spring JMS support? Select one or more correct answers.

1. By implementing the javax.jms.MessageListener interface
2. By implementing the SessionAwareMessageListener interface provided by Spring
3. Without any code, only using the jms namespace provided by Spring
4. By writing a single POJO without parent class or interface

JMX

Question 49

What is easier to do by using Spring JMS support? Select one or more correct answers.

1. Register any Spring bean as JMX MBean
2. Register an existing MBean with a MBeanServer
3. Accessing to remote MBean
4. Control the attributes and the operations of a Spring bean exposes as a MBean

Question 50

What is the purpose of the @ManageResource annotation? Select a single answer.

1. Expose a bean's property (getter/setter) to JMX
2. Expose a bean's method to JMX
3. Identify a Spring bean as a JMX MBean
4. None of the above

Response

Container

Question 1

Answer 2 is correct. Those beans are anonymous because no id is supplied explicitly. Thus Spring container generates a unique id for that bean. It uses the fully qualified class name and appends a number to them. However, if you want to refer to that bean by name, through the use of the `ref` element you must provide a name (see Naming Beans section of the Spring reference manual). To be correct, the 2nd bean has to declare a `jpaDao` id attribute in order to be reference by the `repository` property of the first bean.

Question 2

Answers 1 and 4 are correct.

1. To set bean's property with the p:propertyName shortcut, you have to declare the http://www.springframework.org/schema/p in your xml configuration file. No xsd is required.
2. The bean is anonymous. Spring generates a unique id: `com.spring.service.BankServiceImpl#0`
3. To reference another bean with the p namespace, you have to use the p:propertyName-ref syntax
4. Due to the above explanation, NationalBank is not a bean reference, so it is a simple String and thus a scalar value.

Question 3

Correct answer is 3.
The @Bean annotation defines a String bean with the id "clientRepository". JpaClientRepository is the implementation class of the bean. The data source is injected and is not declared in this class.

Question 4

The only possible answer is the number 3.

1. The <util:constant static-field="constant name"/> tag enables to reference a Java constant or enumeration into a spring configuration file
2. ConstantPlaceholderConfigurer does not exist. You may think about the PropertyPlaceholderConfigurer bean post processor.
3. The <context:property-placeholder location="file:/myApp.properties" /> tag activates the replacement of ${...} placeholders, resolved against the specified properties file.
4. The c: namespace is for simplifying constructor syntax (since Spring 3.1) and don't provide such feature.

Question 5

The statements number 5 is right.

1. You may auto-wiring properties by constructor, setter or properties in the same bean
2. The <constructor-arg> tag helps to instanciated a bean without default or no-args constructor
3. The <constructor-arg> tag could take type and index to reduce ambiguity, but not name which requires debug symbols.

Question 6

Answers 1, 3, 4 and 5 are rights.

1. The @PostConstruct, @PreDestroy and @Resource annotations are defined in the JSR-250
2. They belong to the javax.annotation package. You should add an external jar to use them in Java 5. Java 6 and 7 integrates them.
3. The <context:component-scan> automatically detects stereotyped classes and turns on the <context:annotation-config>
4. The <context:annotation-config > activates the Spring infrastructure for various annotations to be detected in bean classes, including the JSR 250 annotations
5. The CommonAnnotationBeanPostProcessor supports common Java annotations out of the box, in particular the JSR-250 annotations.

Question 7

Correct answer in the number 3.

1. In a web application, the ContextLoaderListener is in charge to create an WebApplicationContext.
2. In an integration test based on Spring, the SpringJUnit4ClassRunner creates the application context for you. The @ContextConfiguration annotation allows to specified application context configuration files.
3. In a main method, you have to instanciated a class implementing the ApplicationContext interface (examples: ClassPathXmlApplicationContext or FileSystemXmlApplicationContext)

Question 8

Answer number 4 is right.

1. `When using the ClassPathXmlApplicationContext,` the classpath: prefix is default one so you could omit it
2. In a Spring location resource, package separator is a slash and not a dot. Thus the com/example/myapp/config.xml syntax has to be used.
3. ClassPathXmlApplicationContext starts looking from root of the classpath regardless of whether specify "/"

Question 9

Answers number 3 and 4 are valid.

1. The @Autowired annotation has no name property, just a required one.
2. Autowiring a field, the @Inject or the @Autowired or the @Resource annotations are mandatory.
3. The @Qualifier("name") annotation complements the use of the @Autowired annotation by specifying the name of the bean to inject
4. When 2 beans are eligible to auto-injection, Spring uses the field name to select the appropriate one.

Question 10

Answers number 1 and 4 are valid.

1. With modern mock API like Mockito or EasyMock, interfaces are not mandatory for mocking or stubbing the service. But using interface remains easier when you have to manually mock the service in unit test.
2. Auto-injection is possible with class. Spring uses CGLIB.
3. Dependency checking is an advantage of dependencies injection.
4. The Inversion of Control pattern requires an interface to separate 2 classes. This pattern provides code more flexible, unit testable, loosely coupled and maintainable.

Question 11

Correct answers: 4

1. In the bean lifecycle, method annotated with @PostConstruct is called after the properties set step and the BeanPostProcessors#postProcessBeforeInitialization step
2. Destroy methods of prototype beans are never called
3. In the bean lifecycle, the afterPropertiesSet callback method of the InitializingBean is called after the method annotated with the @PostConstruct annotation and before the init-method declared in the XML configuration file.
4. In the bean lifecycle, the method annotated with the @PreDestroy annotation is called before the destroy callback of the DisposableBean interface and before the destroy-method declared in the XML configuration file.

Question 12

Correct answers are 1 and 2.

1. In order to be taken into account by Spring, the ApplicationConfig class has to be annotated with the @Configuration annotation
2. Default or no-arg constructor is mandatory. Here, the provided constructor with a dataSource parameter is not taken into account
3. The bean name is clientRepository. The name property of the @Bean annotation is specified thus the method name jpaClientRepository is ignored.
4.

Question 13

Correct answers are 3 and 4

1. Use <tx:annotation-driven /> to enable @Transactional annotation scanning
2. Use <aop:aspectj-autoproxy /> to enable detection of @Aspect bean
3. Turns on <context:annotation-config /> or <context:component-scan /> to enable @Autowiring annotation
4. Turns on <context:component-scan /> to enable @Component annotation scanning

Test

Question 14

The only correct answer is number 3.

1. The Spring context is cached across tests unless you use @DirtiesContext annotation
2. With the Spring test module, dependency injection is available in test case. So you may auto-wired the bean to test
3. By default, a @ContextConfiguration annoted class inherits the spring context configuration file locations defined by an annotated superclass. The inheritLocations of this attribute allows to change this default behavior.
4. If no context configuration file is provided to the @ContextConfiguration annotation, Spring use a file convention naming. It try to load a file named with the test class name and suffices by the "-context.xml" suffice (i.e. MyDaoTest-context.xml)

Question 15

Correct answers are 1, 3 and 4.

What are the main advantage(s) for using Spring when writing integration tests?

1. More than testing multiple classes together, integration test may allow to test your spring configuration file and/or to reuse it.
2. Mocking or stubbing is more frequent in unit tests than in integration tests. And Spring does not provide any implementation or abstraction of mock framework.
3. The framework may create and roll back a transaction for each test method. Default rollback policy could be change by using the @TransactionConfiguration annotation. And default mode could be overridden by the @Rollback annotation.
4. DependencyInjectionTestExecutionListener provides support for dependency injection and initialization of test instances.

Question 16

The correct answer is the number 3.

What are the main advantage(s) for using Spring when writing unit tests?

1. You don't need Spring container to writer unit test
2. Refer to the answer number 1.
3. The org.springframework.mock package provides mock classes like MockHttpSession or MockHttpContext. They could be helpful for unit test in the presentation layer and when you don't use any mock framework such as Mockity or EasyMock.

Question 17

Answer 5 is correct.

What is right about the spring test module?

1. The spring test module does not provide an abstraction layer for open source mock frameworks like EasyMock, JMock or Mockito
2. The @Mock annotations comes from the Mockito framework
3. The spring test module does not provide mechanism to generate mock objects at runtime

Question 18

Correct statements are number 1 and 4.

1. The transactionManager property of the @TransactionConfiguration annotation enable to set the bean name of the PlatformTransactionManager that is to be used to drive transactions.
2. Method annotated with @Before is executed inside the test's transaction. You have to use the @BeforeTransaction to execute code outside the test's transaction.
3. The REQUIRES_NEW propagation suspends the current test's transaction then creates a new transaction that will be used to execute the service. A commit at the service level could not be changed by the test.
4. The transaction for the annotated method should be rolled back after the method has completed.

AOP

Question 19

The correct answer is the number 4.

Considering 2 classes AccountServiceImpl and ClientServiceImpl. Any of these 2 classes inherits from each other. What is the result of the pointcut expressions?

```
    execution(* *..AccountServiceImpl.update(..))
&& execution(* *..ClientServiceImpl.update(..))
```

Poincut expression could not satisfied both first and second execution point. Do not confuse the && operator and || operator.

Question 20

Correct answer is the number 4.

Due to the proxy-based nature of Spring's AOP framework, protected methods are by definition not intercepted, neither for JDK proxie nor for CGLIB proxies. As a consequence, any given pointcut will be matched against public methods only!

To intercept private and protected methods, AspecJ weaving should be used instead of the Spring's proxy-bases AOP framework.

Question 21

The 2 correct statements are 1 and 5.

What are the 2 correct statements about AOP proxy.

1. An object created by the AOP framework in order to implement the aspect contracts
2. If the target object does not implement any interfaces then a CGLIB proxy will be created. You could also use CGLIB proxy instead of JDK dynamic proxy
3. If the target object does not implement any interfaces then a CGLIB proxy will be created.
4. When CGLIB proxy is used, final methods cannot be advised, as they cannot be overridden.
5. AOP Proxies are created by the AbstractAutoProxyCreator#postProcessAfterInitialization method.

Question 22

The answer number 2 is correct.

1. A before advice could throw an exception
2. An after throwing advice is executed if a method exits by throwing an exception
3. An advice that executes before a join point is named a before advice
4. Spring supports after throwing advices

Question 23

Correct answer: 4

1. This is an after (finally) advice
2. This is an around advice
3. This is a before advice
4. True

Question 24

Correct answer: 1

1. Definition of an advice
2. Definition of a joint point
3. Represents nothing
4. Definition of a point cut

Question 25

Correct answer: 2

1. Definition of an advice
2. Definition of a pointcut
3. Represents nothing

Question 26

Correct answers: 1, 3

Select methods that match with the following pointcut:

execution(* com.test.service..*.*(*))

1. True
2. The pattern (*) matches a method taking one parameter of any type
3. The com.test.service.account sub-package matches the pointcut
4. False for the same reason as answer number 2.

Question 27

Correct answers: 2

1. Interception of constructors requires the use of Spring-driven native AspectJ weaving instead of Spring's proxy-based AOP framework
2. The @annotation designator enables to select methods that are annotated by a given annotation
3. The staticinitialization AspectJ designator is not supported by Spring AOP
4. Pointcut expressions can be combined using &&, || and !

Question 28

Correct answers: 1

1. The execution of all public method
2. The * return type pattern indicates any return value or void
3. The (..) param pattern indicates 0, 1 or many parameters

4. No package name is specified. So classes of any package could match.

Data Access

Question 29

Correct answers: 2 , 4

1. Spring exception translation mechanism has nothing to do with read-only transaction
2. Read-only transaction prevents Hibernate from flushing its session. Hibernate do not do dirty checking and it increases its performance.
3. No
4. When jdbc transaction is marked as read-only, Oracle only accepts SELECT SQL statements.

Question 30

Correct answers: 1, 3, 4

1. JDBC is supported: JdbcTemplate, JDBCException wrapper …
2. Some NoSQL databases are supports through the Spring Data project
3. Hibernate is supported: HibernateTemplate, AnnotationSessionFactoryBean …
4. JPA is supported: LocalEntityManagerFactoryBean, @PersistenceContext annotation support

Question 31

Correct answer: 1

1. A JdbcTemplate requires a DataSource as input parameters
2. JdbcTemplate uses the probided datasource to open then close a JDBC connection
3. Callback methods of JdbcTemplate throw SQL Exception and Spring converts into DataAccessException
4. For example, the queryForInt method executes an SQL statement

Question 32

Correct answer: 3

1. RowMapper : result set parsing when need to map each row into a custom object
2. RowCallbackHandler : result set parsing without returning a result to the JdbcTemplate caller
3. ResultSetExtractor : for result set parsing and merging rows into a single object
4. ResultSetMapper : this class does not exist

Question 33

Correct answer: 3

1. False. This is supported by the AnnotationSessionFactoryBean using annotedClasses
2. False. This is supported by the AnnotationSessionFactoryBean using packagesToScan
3. True using mappingLocations
4. False

Transaction

Question 34

Correct answer: 2

1. <tx:jta-transaction-manager />
2. Id of the transaction manager bean could be customized (ie. txManager)
3. DataSourceTransactionManager is a transaction manager for a JDBC data source. HibernateTransactionManager may be used to manage transaction with Hibernate.
4. The AbstractPlatformTransactionManager has a defaultTimeout property that could be customized

Question 35

Correct answer: 2

In proxy mode, only external method calls coming in through the proxy are intercepted. In the code snippet, the add() method is self-invocated. This means that, the @Transactional annotation of the add() method is not interpreted. The REQUIRES_NEW propagation level is not taken into account. To summary, when the transferMoney() methods calls add() method directly, the transaction attributes of add() method are not used

Question 36

Correct answer: 1

1. The TransactionTemplate class provides an execute(TransactionCallback) method
2. The TransactionService class does not exists
3. The @Transactional annotation is for declarative transaction management

Question 37

Correct answer: 3

1. PROPAGATION_REQUIRED
2. PROPAGATION_NESTED
3. PROPAGATION_REQUIRES_NEW

Question 38

Correct answer: 2

1. False.
2. True
3. False
4. False

Sping @MVC

Question 39

Correct answer: 1

1. Spring does not allow to return an absolute path to the view
2. Controller could return a String that matches with a logical view name
3. A JstlView with the .jsp path (i.e. /WEB-INF/accountList.jsp)
4. void forward to the default view
5. null forward to the default view

Question 40

Correct answer: 2, 4

1. Spring MVC controllers are beans. So you can declare them into a Spring application context XML configuration file that could be loaded by the DispatcherServlet.
2. In the web.xml, you may declarer and a ContextLoaderListener and a DispatcherServlet that are in charge to load XML Spring configuration files. But you cannot declare controllers directly in these file.
3. The @Controller annotation may be used to annoted Spring MVC Controller beans that handle HTTP requests.
4. JSP is the View of the MVC Pattern. Thus this is not the right place to declare controllers.

Question 41

Correct answer: 3

1. HttpServletRequest and HttpServletResponse have to be mocked. Id of the account to display could be set into the http request parameters.

2. HttpServletRequest and HttpSession have to be mocked. Id of the account to display could be set into the http request parameters.

3. This method is not dependent of the servlet API. Id of the account to display may be directly passed through the call stack. Thus test methods are simplified.

4. The *@PathVariable* annotation has to be bound to a URI template variable. This is not the case.

Sping Security

Question 42

Correct answer: 4

1. @Secured annotation is a Spring Security annotation
2. @RolesAllowed is a JSR-250 annotation that is supported by Spring Security
3. Spring Security could be configured in a XML way to intercept particular URLs

Question 43

Correct answer: 1, 2, 3 and 4

Question 44

Correct answer: 2

1. You cannot mix EL and constant in the same configuration file
2. If more than one intercept-url matches, the top one is used
3. Ant pattern is used by default. But you can change to use regular expression.
4. Security rules may apply to request URL, request method (GET, POST …) but not to request parameters.

Remoting

Question 45

Correct answer: 4

1. No more interface to implement. RMI Client and Server could be POJO.
2. No more class to extend. RMI Client and Server could be POJO.
3. Spring Remoting wraps the cheched RemoteException into RuntimeExcpetion.
4. Object that are transferred via RMI are serializabled/unserializabled. So they have to implement the Serializable interface.

Question 46

Correct answers: 1, 3

1. HttpInvokerServiceExporter requires a HTTP web server to process incoming http request. Tomcat or Jetty is possible candidates. Spring also supports the Oracle/Sun's JRE 1.6 HTTP server.
2. Only the POST method is supported. Maybe due to the 256 characters limit of the GET method.
3. Spring comes with 2 http client implementations: for Commons HttpClient and classic JavaSE API. You can create a custom one by extending the AbstractHttpInvokerRequestExecutor class.
4. Does not support SOAP web service. Use the Spring web service module or use the JAX-WS or JAX-RPC remoting support.

JMS

Question 47

Correct answer: 2

1. The convertAndSend method sends a given object to a destination, converting the object to a JMS message.
2. The onMessage method does not exist.
3. The receiveAndConvert method receives a message synchronously then convert the message into an object
4. The setDefaultDestination method sets the destination to be used on send/receive operations that do not have a destination parameter.

Question 48

Correct answers: 1, 2 , 4

1. The javax.jms.MessageListener interface could be used with the SimpleMessageListenerContainer
2. The SessionAwareMessageListener interface could be used with DefaultMessageListenerContainer and SimpleMessageListenerContainer
3. Business code is required to handle and process the JMS message.
4. A JMS Listener could be a POJO. The name of the handler method to invoke has to be specified in the <jms:listener /> tag.

JMX

Question 49

Correct answers: 1, 2, 3, 4

1. The MBeanExporter class allow to expose any Spring bean as a JMX MBean
2. Exiting MBean could be declared as Spring bean. Then the <context:mbean-export /> directive enables their registration to the MBeanServer

3. Remote MBean could be access through a proxy
4. Implementations of the MBeanInfoAssembler interface do the job

Question 50

Correct answer: 3

1. @ManageAttribute exposes a bean's property (getter/setter) to JMX
2. @ManageOperation exposes a bean's method to JMX
3. @ManageResources identify a Spring bean as a JMX MBean

Question 1. URLConnection instance represents a link for accessing or communicating with the resource at the location?
Your Answer: true
✓ Correct Answer

Question 2. JSP technology is extensible?
Your Answer: true
✓ Correct Answer

Question 3. What tools can be used to generate the multiple views in jsp?
Your Answer: xalan-J
✓ Correct Answer

Question 4. What is a Scriptlet in JSP?
Your Answer: <% %>
✓ Correct Answer

Question 5. The maximum age of the cookie in JSP can be set by?
Your Answer: cookie.setMaxAge(int seconds)
✓ Correct Answer

Question 6. What is the possible way of communication between applet and Servlet?
Your Answer: All of the above
✓ Correct Answer

Question 7. Which of the scripting of JSP not putting content into service method of the converted servlet?
Your Answer: Declarations
✓ Correct Answer

Question 8. JSP stands for __ ?
Your Answer: Java Server Pages
✓ Correct Answer

Question 9. The Page directive in JSP is defined as follows: Then which of the implicit objects won't be available ?
Your Answer: session, exception
✓ Correct Answer

Question 10. Can we implement an interface in a JSP?
Your Answer: No
✓ Correct Answer

Question 11. config is an implicit object in jsp.
Your Answer: TRUE
✓ Correct Answer

Question 12. What is a Declaration in jsp?

Your Answer: <%! %>
✓ Correct Answer

Question 13. All the _____ data is kept at the application server _____ data is kept at the web server?
Your Answer: dynamic, static
✓ Correct Answer

Question 14. What is a translation unit?
Your Answer: When the JSP engine is presented with such a JSP page it is converted to one Servlet class and this is called a translation unit
✓ Correct Answer

Question 15. Which of the following statements are true about GET and POST methods?
Your Answer: In GET, parameters are appended to the request line
✓ Correct Answer

Question 16. The jsp:plugin tag is used to insert the browser-specific OBJECTS and EMBED elements?
Your Answer: true
✓ Correct Answer

Question 17. Fill in the blanks -- Sharing of session data across multiple load balanaced web servers is called _____ ?
Your Answer: Clustering
✓ Correct Answer

Question 18. The response.sendRedirect("..."); is the response implicit object to redirect the browser to the different resource?
Your Answer: true
✓ Correct Answer

Question 19. URL encoding is the method of replacing all the spaces and other extra characters into their corresponding _____ Characters?
Your Answer: Hex
✓ Correct Answer

Question 20. The < % return; % > simply aborts the processing of JSP?
Your Answer: true
✓ Correct Answer

Question 21. The error message displayed when the page is not found at the correct location?
Your Answer: 404
✓ Correct Answer

Question 22. The JspPage interface defines the _____ and ____ method which the page writer can use in their pages and are invoked in much the same manner as the ____ and _____ methods of a servlet.

Your Answer: jspInit(), jspDestroy(), init() , destroy()
✓ Correct Answer

Question 23. To show money format on the JSP page we use _____ class?
Your Answer: NumberFormat
✓ Correct Answer

Question 24. Which of the following statements is true regarding the scope of 'request' in JSP?
Your Answer: Objects with request scope are accessible from pages processing the same request where they were created.
✓ Correct Answer

Question 25. Which of the following are the implicit objects in JSP?
Your Answer: request, response, session
✓ Correct Answer

Question 26. If the session object is invalidated can it be again regained or refreshed?
Your Answer: No
✓ Correct Answer

Question 27. Can we implement an interface in JSP?
Your Answer: false
✓ Correct Answer

Question 28. The browsers are only required to accept_____ cookies per site?
Your Answer: 20
✓ Correct Answer

Question 29. Which tag in the jsp is used to define the error page?
Your Answer: <%@page isErrorPage="true" %>
✓ Correct Answer

Question 30. pageContext is an implicit object in jsp.
Your Answer: TRUE
✓ Correct Answer

Question 31. The listener is notified when the Servlet context (i.e., the Web application) is initialized and destroyed?
Your Answer: Servlet context listeners
✓ Correct Answer

Question 32. The_____ listener is notified when attributes are added to, removed from, or replaced in the Servlet context?
Your Answer: Servlet context attribute
✓ Correct Answer

Question 33. Given following form:
<form action="register.do">

<input type="text" name="Name">
<input type="submit" value="Save">
</form>

and a servlet code:
public class RegisterServlet extends HttpServlet{
pulic void doPost(HttpServletRequest req, HttpServletResponse res) {
// registration logic goes here
return;
}
}

With the above code, assuming the servlet is configured properly and registration logic works good, trying to register user fails.
Choose one reason.

Your Answer: The registration fails because, above servlet dont have doGet() method.
✓ Correct Answer

Question 34. What is a Hidden Comment in JSP?
Your Answer: <%-- --%>
✓ Correct Answer

Question 35. What is a Expression in JSP?
Your Answer: <%= %>
✓ Correct Answer

Question 36. What is the result of adding following header?
response.addDateHeader("Expires",-1);
Your Answer: To ensure the page does not get cached at the client browsers.
✓ Correct Answer

Question 37. The maximum length of the session Id(length identifier) is _____?
Your Answer: 4k
✓ Correct Answer

Question 38. Which methods can't be overridden in the JSP page?
Your Answer: _jspService()
✓ Correct Answer

Question 39. Can a JSP process HTML form data?
Your Answer: true
✓ Correct Answer

Question No :1
Which methods can't be overridden in the JSP page ?

(Choose correct one from multiple below)
1. jspDestroy()
2. jspInit()
3. _jspService()
4. getParameter()
correct is :3

Question No :2
The tag %@include file=?.....? % > helps in including ?

(Choose correct one from multiple below)
1. readOnly
2. dynamic
3. static
4. None of the above
correct is :3

Question No :3
The jsp:useBean attribute is used to indicate the Serialized bean?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :4
Is data in request object avilable after response.sendRedirect();
(Choose correct one from multiple below)
1. yes avilable
2. Not avilable
3. Can't say
4. None of the above
correct is :2

Question No :5
The response.sendRedirect("??..?); is the response implicit object to redirect the browser to the different resource?

(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :6
The jsp:plugin tag is used to insert the browser-specific OBJECTS and EMBED elements?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :7
Which tag in the jsp is used to define the error page?
(Choose correct one from multiple below)
1. <%@page isErrorPage="yes" %>
2. <%@page isErrorPage="true" %>
3. <%@page isErrorPage="false" %>
4. None of the above
correct is :2

Question No :8
Can we make use of a ServletOutputStream object from the JSP page?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. None of the above
correct is :1

Question No :9
Adding a meta, usage like <meta name=?das? contents=?no index?> tag to the jsp prevents it from being indexed by the search engines like Yahoo and Google

(Choose correct one from multiple below)
1. true
2. false
3. can't say

4. none of the above
correct is :1

Question No :10
To show money format on the JSP page we use _____ class ?

(Choose correct one from multiple below)
1. NumberFormat
2. StringBuffer
3. CurrencyFormat
4. AirthmaticFormat
correct is :1

Question No :11
The error message displayed when the page is not found at the correct location?
(Choose correct one from multiple below)
1. 500
2. 404
3. 505
4. 440
correct is :2

Question No :12
Request.getServerName () is used to get the name of the server on which the Jsp is running?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :13
Which of the following statements is true about JSP tag library?

(Choose correct one from multiple below)
1. It defines the standard tag that works the same everywhere
2. It is a single library and we can use it in multiple jsp containers

3. It has support for the common structural tasks like iteration and condition.
4. All of the above
correct is :4

Question No :14
Which of the scripting of JSP not putting content into service method of the converted servlet ?

(Choose correct one from multiple below)
1. Declarations
2. Scriptlets
3. expressions
4. None of the above
correct is :1

Question No :15
Can we implement an interface in JSP ?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :2

Question No :16
ServletContext class gives information about the _____?
(Choose correct one from multiple below)
1. request
2. session
3. container
4. none of the above
correct is :3

Question No :17
PageContext class gives information about the _____?
(Choose correct one from multiple below)
1. application
2. session
3. request
4. response
correct is :3

Question No :18
While using request.getRequestDispatcher(path) we need to give the _____ path of the object ?
(Choose correct one from multiple below)
1. absolute
2. relative
3. can't say
4. none of the above
correct is :2

Question No :19
While using context.getRequestDispatcher(path) we need to give the _____ path of the object ?
(Choose correct one from multiple below)
1. absolute
2. relative
3. can't say
4. none of the above
correct is :1

Question No :20
URL instance represents the location of a resource?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :21
URLConnection instance represents a link for accessing or communicating with the resource at the location?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :22
When u try to redirect a page after you already have written something in your page, the error encountered is
?Response has already been committed error? or popularly called 402 error.?
(Choose correct one from multiple below)

1. true
2. false
3. none of the above
4. none of the above
correct is :1

Question No :23
The wrapper function like
<%!
String blanknull(String s) {
return (s == null) ? "" : s;
}
%>
then use it inside your JSP form, like
<input type="text" name="size"
value="<%=blanknull(size)% >" >
prevents the word_____ from apperaring in an HTML page.

(Choose correct one from multiple below)
1. Null
2. Not Null
3. space
4. None of the above
correct is :1

Question No :24
The code block
<%@ page isErrorPage="true" %>
<%
out.println(" ");
PrintWriter pw = response.getWriter();
exception.printStackTrace(pw);
out.println(" ");
%>
Helps in printing the _____ of exception on JSP.
(Choose correct one from multiple below)
1. stacktrace
2. message
3. error message
4. none of the above
correct is :1

Question No :25
The code block
<%@ page isErrorPage="true" %> is used for ?

(Choose correct one from multiple below)
1. print error
2. print output
3. get input
4. none of the above
correct is :1

Question No :26
Which of the following correctly defines JSP technology ?

(Choose correct one from multiple below)
1. JSP page is a text-based document that describes how to process a request to create a response.
2. JSP page is a text-based document that describes how to process a to response create a request.
3. JSP page is a xml-based document that describes how to process a request to create a response.
4. JSP page is a xml-based document that describes how to process a to response create a request.
correct is :1

Question No :27
URL encoding is the method of replacing all the spaces and other extra characters into their corresponding _____ Characters ?

(Choose correct one from multiple below)
1. Hex
2. Binary
3. Octal
4. Decimal
correct is :1

Question No :28
All the _____ data is kept at the application server _____ data is kept at the web server ?

(Choose correct one from multiple below)
1. dynamic, static
2. static, dynamic
3. HTML, Servlet
4. Servlet, HTML
correct is :1

Question No :29
Under JSP 1.0 , the implicit object which can be used for reference is ?this?. It is used to refer to the _____ ?

(Choose correct one from multiple below)
1. Servlet generated by the JSP page
2. Calling the previous JSP page
3. Servlet called by the jsp page
4. None of the above
correct is :1

Question No :30
Which of the following are the implicit objects in JSP
1. Application, out
2. config, exception
3. page, pageContext
4. request, response, session

(Choose correct one from multiple below)
1. 1, 2
2. 2, 3
3. 2, 3, 4
4. 1, 2, 3, 4
correct is :4

Question No :31
Can a JSP process HTML form data ?
(Choose correct one from multiple below)
1. true

2. false
3. can't say
4. none of the above
correct is :1

Question No :32
State true or false -- It is possible to call an external application like MSWord by the click on the JSP page or Servlet?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :33
What tools can be used to generate the multiple views in jsp ?

(Choose correct one from multiple below)
1. xalan-J
2. Saxon
3. Jdk1.5
4. None of the above
correct is :1

Question No :34
What is the possible way of communication between applet and Servlet ?

(Choose correct one from multiple below)
1. HTTP Communication (Text-based and object-based)
2. Socket Communication
3. RMI Communication
4. All of the above
correct is :4

Question No :35
JSP technology is extensible?
(Choose correct one from multiple below)

1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :36
What is a translation unit ?

(Choose correct one from multiple below)
1. JSP page can include the contents of other HTML pages or other JSP files.
2. When the JSP engine is presented with such a JSP page it is converted to one Servlet class and this is called a translation unit,
3. In a translation unit is that page directives affect the whole unit, one variable declaration cannot occur in the same unit more than once, the standard action jsp:useBean cannot declare the same bean twice in one unit.
4. All of the above
correct is :2

Question No :37
HttpSession.setMaxInactiveInterval() can manage the inactivity lease period on a per-session basis?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :38
Some Operating Systems have a limitation on the Max Length of the URL, which are normally _____ characters?
(Choose correct one from multiple below)
1. 566
2. 260
3. 256
4. 255
correct is :4

Question No :39
Fill in the blanks -- Sharing of session data across multiple load balanaced web servers is called

_____ ?
(Choose correct one from multiple below)
1. Clustering
2. ShareWeb
3. loadBalancer
4. None of the above
correct is :1

Question No :40
The session tracking in the JSP can be done by
1. URL rewriting
2. Cookies
3. User-Authorization
4. Hidden value

(Choose correct one from multiple below)
1. 1,3,4 only
2. 2,3,4 only
3. 1,2,4 only
4. 1,2,3,4
correct is :3

Question No :41
From the current Browser window, if we open a new Window, then it referred to as Multiple Windows. Are Sessions properties are maintained across all these windows, even though they are operating in multiple windows ?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :42
If the session object is invalidated can it be again regained or refreshed?
(Choose correct one from multiple below)
1. Yes
2. No
3. Can't say
4. none of the above

correct is :2

Question No :43

What is the role played by JSP in the MVC architecture ?

(Choose correct one from multiple below)

1. Model
2. View
3. Controller
4. None of the above

correct is :2

Question No :44

The listener is notified when the Servlet context (i.e., the Web application) is initialized and destroyed ?

(Choose correct one from multiple below)

1. Servlet context attribute
2. Servlet context listeners
3. Session Attribute
4. Session Listeners

correct is :2

Question No :45

The_____ listener is notified when attributes are added to, removed from, or replaced in the Servlet context?

(Choose correct one from multiple below)

1. Session context Listeners
2. Servlet context attribute
3. Session Listeners
4. Session Attribute

correct is :2

Question No :46

The _____ listeners are notified when session objects are created, invalidated, or timed out.

(Choose correct one from multiple below)

1. Servlet context listeners

2. Session Listeners
3. Servlet context attribute
4. Session Attribute
correct is :2

Question No :47
Arrange the lifecycle phases of JSP in correct order1. Page translation: -page is parsed, and a java file which is a Servlet is created. 2. Page compilation: page is compiled into a class file 3. Page loading : This class file is loaded. 4. Create an instance :- Instance of Servlet is created 5. jspInit() method is called 6. _jspService is called to handle service calls 7. _jspDestroy is called to destroy it when the Servlet is not required.

(Choose correct one from multiple below)
1. 4,5,3,6,2,7,1
2. 1,2,3,4,5,6,7
3. 1,4,3,2,5,6,7
4. 3,2,1,4,5,6,7
correct is :2

Question No :48
The JspPage interface defines the _____ and ____ method which the page writer can use in their pages and are invoked in much the same manner as the ____ and _____ methods of a servlet.

(Choose correct one from multiple below)
1. jspInit(), jspDestroy(), init() , destroy()
2. init(),jspDestroy(),jspInit(),destroy()
3. destroy(),jspDestroy(),jspInit(),init()
4. init(),destroy(),jspInit(), jspDestroy()
correct is :1

Question No :49
Which of the following statements is true regarding the scope of ?request? in JSP ?

(Choose correct one from multiple below)

1. Objects with request scope are accessible from pages processing the same request where they were created.
2. All references to the object shall be released after the request is processed; in particular, if the request is forwarded to a resource in the same runtime, the object is still reachable.
3. References to objects with request scope are stored in the request object.
4. All of the above.
correct is :1

Question No :50
There is a limit to the max amount of data that can be stored in the session object ?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :2

Question No :51
The maximum length of the session Id(length identifier) is _____?
(Choose correct one from multiple below)
1. 4k
2. 2k
3. 6k
4. 8k
correct is :1

Question No :52
The browsers are only required to accept_____ cookies per site?
(Choose correct one from multiple below)
1. 20
2. 30
3. 22
4. 24
correct is :1

Question No :53
The maximum age of the cookie in JSP can be set by ?

(Choose correct one from multiple below)
1. cookie.setAgeMax (int seconds)
2. cookie.setAgeMax (float seconds)

3. cookie.setMaxAge(float seconds)
4. cookie.setMaxAge(int seconds)
correct is :4

Question No :54
The ___ indicates that the cookie will expire once the browser is closed?
(Choose correct one from multiple below)
1. cookie.setMaxAge(-1)
2. cookie.setMaxAge(0)
3. cookie.setMaxAge(365)
4. None of the above
correct is :1

Question No :55
The ____ is used to delete a cookie?
(Choose correct one from multiple below)
1. cookie.setMaxAge(0)
2. cookie.setMaxAge(-1)
3. cookie.setMaxAge(-2)
4. None of the above
correct is :1

Question No :56
Which of the following correctly describes the sequence of calling the methods for preventing browser from caching the details. i.response.setHeader("Cache-Control","no-store");
ii. response.setHeader("Pragma","no-cache"); iii.response.setDateHeader ("Expires", 0);
(Choose correct one from multiple below)
1. i,ii,iii
2. ii,i,iii
3. iii,ii,i
4. None of the above
correct is :1

Question No :57
The < % return; % > simply aborts the processing of JSP?
(Choose correct one from multiple below)
1. true
2. false
3. can't say
4. none of the above
correct is :1

Question No :58

The _____ is used to call other Servlet or JSP from the current Servlet ?

(Choose correct one from multiple below)

1. RequestDispatcher
2. ResponseDispatcher
3. ServletInvoker
4. None of the above

correct is :1

Question 1. Which of the following statements are true?
Your Answer: The Servlet instance variables are not thread safe.
✓ Correct Answer

Question 2. Name the class that includes the getSession method that is used to get the HttpSession object.
Your Answer: HttpServletRequest
✓ Correct Answer

Question 3. I have a web application on tomcat. Here many servlets open their own separate connection to database. To optimize this application, I decided to ask all servlet developers to use same db connection. I need to initialize this connection, before any servlet / jsp will be accessed. I got following suggestions from my developers.

Which is the best way to do it?
Your Answer: Make the connection in a context listener and save it.
✓ Correct Answer

Question 4. Name the http method used to send resources to the server. Select the one correct answer.
Your Answer: PUT method
✓ Correct Answer

Question 5. Which HTTP method gets invoked when a user clicks on a link? Select the one correct answer.
Your Answer: GET method
✓ Correct Answer

Question 6. Which of the following are the methods defined in ServletContextListener interface?
Your Answer: public void contextInitialized(ServletContextEvent event)
✓ Correct Answer

Question 7. Which of the following is not a valid HTTP/1.1 method. Select the one correct answer.
Your Answer: COMPARE method
✓ Correct Answer

Question 8. Is the following statement true or false. URL rewriting may be used when a browser is disabled. In URL encoding the session id is included as part of the URL.
Your Answer: True
✓ Correct Answer

Question 9. Which of the following are correct statements? Select the correct answer.
Your Answer: The getRequestDispatcher(String URL) is defined in both ServletContext and HttpServletRequest method
✓ Correct Answer

Question 10. Which of the following statement is correct. Select the one correct answer.

Your Answer: The HttpServletResponse defines constants like SC_NOT_FOUND that may be used as a parameter to setStatus method.
✓ Correct Answer

Question 11. Which of the following method sets application / context init parameter?
Your Answer: Its not possible in the method.
✓ Correct Answer

Question 12. Name the method defined in the HttpServletResponse class that may be used to set the content type. Select the one correct answer.
Your Answer: setContentType
✓ Correct Answer

Question 13. What is the result of following code, given there already exists a header by name "TestHeader".

response.addHeader("TestHeader","TEST VALUE");
Your Answer: The "TEST VALUE" will be appended to the old value of the header.
✓ Correct Answer

Question 14. Which HTTP method is used by the client to check what server receives when request is made?
Your Answer: TRACE
✓ Correct Answer

Question 15. When using HTML forms which of the folowing is true for POST method? Select the one correct answer.
Your Answer: POST method sends data in the body of the request.
✓ Correct Answer

Question 16. To send text output in a response, the following method of HttpServletResponse may be used to get the appropriate Writer/Stream object. Select the one correct answer.
Your Answer: getWriter
✓ Correct Answer

Question 17. I have an attribute class

Class User{
String name;
String address;
String phoneNo;
boolean isLogged;
}
This class will be added to the session when user
logs in and removed from the session when user logs out.
I want its isLogged flag to be changed when user login and logs out.

Using which listener I can do it?
Your Answer: javax.servlet.http.HttpSessionBindingListener

✓ Correct Answer

Question 18. In the init(ServletConfig) method of Servlet life cycle, what method can be used to access the ServletConfig object?
Your Answer: getServletConfig()
✓ Correct Answer

Question 19. The sendRedirect method defined in the HttpServlet class is equivalent to invoking the setStatus method with the following parameter and a Location header in the URL. Select the one correct answer.
Your Answer: SC_MOVED_TEMPORARILY
✓ Correct Answer

Question 20. The sendError method defined in the HttpServlet class is equivalent to invoking the setStatus method with the following parameter. Select the one correct answer.
Your Answer: SC_NOT_FOUND
✓ Correct Answer

Question 21. A user types the URL http://www.withoutbook.com/Blog.php . Which HTTP request gets generated. Select the one correct answer.
Your Answer: GET method
✓ Correct Answer

Question 22. Which of the following statements are true about setHeader() and addHeader() methods?
Your Answer: addHeader() adds an additional value to an existing header
✓ Correct Answer

Question 23. My Application has a servlet, which do addition as shown below and sets the value in context scope. It then prints the result.

```
public void doGet(HttpServletRequest req, HttpServletResponse resp) {
PrintWriter out=response.getWriter();
param_1 = request.getParameter("Param1");
param_2 = request.getParameter("Param2");
getServletContext.setAttribute("result",(param_1 + param_2));
response.
out.print("Result =="+getServletContext.getAttribute("result"));
}
```

But sometimes I get wrong results.How can I fix it?
Your Answer: By Putting the "result" attribute in request scope.
✓ Correct Answer

Question 24. My application has a servlet, which do addition as shown on given values and sets the value in context scope. It then prints the result.

```
PrintWriter out=response.getWriter();
param_1 = request.getParameter("Param1");
```

param_2 = request.getParameter("Param2");
getServletContext.setAttribute("result",(param_1 + param_2));
response.
out.print("Result =="+getServletContext.getAttribute("result"));

But sometimes I get wrong results. What do you think might be the reason?
Your Answer: Many clients must be accessing the servlet
✓ Correct Answer

Question 25. Given a web application ShoopingCart with 2 files:
ShoppingCart/customer/ addProductToCart.jsp
ShoppingCart/customer/ showCart.jsp
which of the following code are valid in addProductToCart.jsp to dispatch request to the showCart.jsp?
Your Answer: RequestDispatcher nextView =
request.getRequestDispatcher("/customer/showCart.jsp");
✓ Correct Answer

Question 26. In which of the following cases the request.getAttribute() will be helpful?
Your Answer: If a servlet or jsp is invoked using forward method.
✓ Correct Answer

Question 27. The method getWriter returns an object of type PrintWriter. This class has println methods to generate output. Which of these classes define the getWriter method? Select the one correct answer.
Your Answer: HttpServletResponse
✓ Correct Answer

Question 28. In the web.xml , session configuration is
<session-config>
<session-timeout>0<session-timeout>
</session-config>

What does this means?
Your Answer: Session are never invalidated.
✓ Correct Answer

Question 29. Which of the following methods can be used to get information about parameters applied to entire application?
Your Answer: ServletContext.getInitParameterNames();
✓ Correct Answer

Question 30. Which three digit error codes represent an error in request from client? Select the one correct answer.
Your Answer: Codes starting from 400
✓ Correct Answer

Question 31. In a web application, running in a webserver, who is responsible for creating request and response objects?
Your Answer: Container

✓ Correct Answer

Question 32. To send binary output in a response, the following method of HttpServletResponse may be used to get the appropriate Writer/Stream object. Select the one correct answer.
Your Answer: getOutputStream
✓ Correct Answer

Question 33. Name the location of compiled class files within a war file? Select the one correct answer.
Your Answer: /WEB-INF/classes
✓ Correct Answer

Question 34. Which of the following statements are correct about the status of the Http response. Select the one correct answer.
Your Answer: A status of 200 to 299 signifies that the request was successful.
✓ Correct Answer

Question 35. Name the http method that sends the same response as the request. Select the one correct answer.
Your Answer: TRACE method
✓ Correct Answer