Q1. JSP includes a mechanism for defining …………………………. or custom tags.

A) static attributes

B) local attributes

**C) dynamic attributes**

D) global attributes

**Q2-Which can not be used as the scope when using a JavaBean with JSP?**

**Answer: (A) Response**

(B) Session

(C) Application

(D) Request

Q3-when we use sequence diagram to-

Ans-**Indicates what messages are sent and when**

4) How to separate multiple attributes

Ans- **comma(,)**

Q5) fill in the blank…..

Try

{

String url =class.forname(“com.sql.jdbc.driver”)

catch

{

Exception \_\_\_\_\_\_\_\_\_\_\_{

printstack e;

}

Exception\_\_\_\_\_\_\_\_\_{

printStack e;

}

Answer **ClassNotFoundException and SQLException**

**Q(6**)

public class Bean{

private String str;

Bean(String str ){

this.str = str;

}

public String getStr() {

return str

}

public boolean equals(Object o){

if (!(o instanceof Bean)) {

return false;

}

return ((Bean) o).getStr().equals(str);

}

public int hashCode() {

return 12345;

}

public String toString() {

return str;

}

}

import java.util.HashSet;

public class Test {

public static void main(String ... sss) {

HashSet myMap = new HashSet();

String s1 = new String("das");

String s2 = new String("das");

Bean s3 = new Bean("abcdef");

Bean s4 = new Bean("abcdef");

myMap.add(s1);

myMap.add(s2);

myMap.add(s3);

myMap.add(s4);

System.out.println(myMap);

}

}

Options are

**A.das abcdef**

B.das abcdef das abcdef

C.das das abcdef abcdef

D.das

Answer :

A is the correct answer.

implemented 'equals' and 'hashCode' methods to get unique result in Set.

Q7.How to write an abstract class in class diagram

ANS:-**Italics**

Q8.How to write an Instance in a class diagram

Ans: **Underlined**

Q9.) Behavioral UML diagram

Ans: **Usecase**

Q10.) Set Attribute and Get Attribute are not part of

Ans: (A) **HTTP SERVLET**

(B) SERVLET REQUEST

(C) SERVLET CONFIG

(D) SERVLET CONTEXT

Q11.) How to find total no. of attributes

Ans: **HttpSessionListener**

Q12.) A abstract class which has concrete class

Ans: **Abstract factory interface**

Q13.Preparedstatement execution options are

a. **initialize**

b. started

c. paused

d. stopped

Q14.In javabean/Usebean reference id define?

a. Class name

b. Package

c. **Instance(answer)**

d. None

Q15.) getConnection(- , - , - ) . options were

a**.) <jdbc url> <db username> < db password>**

b.) <ODBC> <ODBC US> <ODBC ?>

Q16.) PreparedStatement . options were a.) MultipleResult b.) SingleResult

Q17.) a statement in SQL query in prepared statement gives multiple result on executing by using which method. a.)executeUpdate() b.) executeQuery() c.) execute().

Q18.) An SQL query will return single result in the result set?

What is the output for the below code ?

public class NameBean {

private String str;

NameBean(String str ){

this.str = str;

}public String toString() {

return str;

}}

import java.util.HashSet;

public class CollClient {

public static void main(String ... sss) {

HashSet myMap = new HashSet();

String s1 = new String("das");

String s2 = new String("das");

NameBean s3 = new NameBean("abcdef");

NameBean s4 = new NameBean("abcdef");

myMap.add(s1);

myMap.add(s2);

myMap.add(s3);

myMap.add(s4);

System.out.println(myMap);

} }

A**) das abcdef**

B ) das das abcdef abcdef

C) das abcdef abcdef

D )abcdef abcdef

Correct answer is : A

Explanations : Need to implement 'equals' and 'hashCode' methods to get unique Set for user defind objects(NameBean).

String object internally implements 'equals' and 'hashCode' methods therefore Set only stored one value.

Q 19//response.getWriter().append("Served at: ").append(request.getContextPath());

request.getSession().setAttribute("key", new X());

request.getSession().setAttribute("key", new X());

request.getSession().setAttribute("key", "x");

request.getSession().removeAttribute("key");

}}

class X implements HttpSessionBindingListener {

public void valueBound(HttpSessionBindingEvent event) {

System.out.print("B");

}

public void valueUnbound(HttpSessionBindingEvent event) {

System.out.print("UB");

}

}

**BBUBUB** is the answer… In options, it will be in option **(B).**

Q20. The ……………………….. package contains classes that help in connecting to a database, sending SQL statements to the database, and processing the query results.

A) connection.sql

B) db.sql

C) pkg.sql

D) **java.sql(ANS)**

Q21 The ………………………….. method executes a simple query and returns a single Result Set object.

A) executeUpdate()

B) **executeQuery() (ANS)**

C) execute()

D) noexecute()

Q22. The ……………………. method executes an SQL statement that may return multiple results.

A) executeUpdate()

B) executeQuery()

C) **execute()(ANS)**

D) noexecute()

Q23. The …………………… object allows you to execute parametrized queries.

A) ResultSet

B) Parametrized

C**) PreparedStatement(ANS)**

D) Condition

Q24 The parameters of the PreparedStatement object are …………………. when the user clicks on the Query button.

A**) initialized (ANS**)

B) started

C) paused

D) stopped

Q25. The …………………. method sets the query parameters of the PreparedStatement Object.

A) putString()

B) insertString()

C) **setString() (ANS)**

D) setToString()

Q26. Connection object can be initialized using the ………………………. method of the Driver Manager class.

A) putConnection()

B) setConnection()

C) Connection()

D) **getConnetion()(ANS**)

Q27.The JDBC-ODBC Bridge driver translates the JDBC API to the ODBC API and used with …….

A) JDBC drivers

**B) ODBC drivers(ANS)**

C) Both A and B

D) None of the above

Q28. The ……………… object provides you with methods to access data from the table.

**A) ResultSet(ANS)**

B) Parametrized

C) TableStatement

D) Condition