**Question 1**

Give the names of any two searching algorithms

**Answer :** Breadth-First search, Depth limited search, Kruskal's algorithms, Prim's algorithms, Grover's algorithms, Dijikistra's algorithms.

**Question 2**

**What is a self Referential structure?**

**Answer :** It is an important characteristic of the data structures used to implement lists , is that they contain, as a member, a reference variable of the same type as the class itself. For this reason, these data structures are frequently called self-referential or recursive data structures.

**Question 3**

What is a postorder traversal of a binary tree?

**Answer :** In a traversal of a binary tree, all the nodes are processed in some way. (For example, they might be printed.) In a postorder traversal, the order of processing is defined by the rule: For each node, the nodes in the left subtree of that node are processed first. Then the nodes in the right subtree are processed. Finally, the node itself is processed.

**Question 4**

Consider the following string:

String hannah = "Did Hannah see bees? Hannah did.";

1. What is the value displayed by the expression hannah.length()? **Answer :**  32.
2. What is the value returned by the method call hannah.charAt(12)? **Answer :**  e.
3. Write an expression that refers to the letter b in the string referred to by hannah. **Answer :** hannah.charAt(15).

**Question 5**

What is the result of trying to compile and run the following code.

public final static void main(String[] args){

double d = 10.0 / -0;

if(d == Double.POSITIVE\_INFINITY)

System.out.println("Positive infinity");

else

System.out.println("Negative infinity");

}

**Answer :** Output Positive infinity. There is no such thing as a positive or negative zero. Hence the result is always positive infinity.

**Question 6**

Is this legal? Why?

long longArr[];

int intArr[] = { 7 ,8 , 9};

longArr = intArr;

**Answer : No.** You cannot assign a reference to an array of primitives to another unless they contain the same primitive types.

**Question 7**

What is the value of d that will be printed out.

public class Test {

public final static void main(String[] args)

{

double d = - 22.22222;

System.out.println(Math.ceil(d));

}

}

**Answer :** -22.0

**Question 8**

How do you create a database connection?

**Answer :** The database connection is created in 3 steps:

1. Find a proper database URL (see FAQ on JDBC URL)

2. Load the database driver

3. Ask the Java DriverManager class to open a connection to your database

In java code, the steps are realized in code as follows:

1. Create a properly formatted JDBR URL for your database. (See FAQ on JDBC URL for more information). A JDBC URL has the form jdbc:someSubProtocol://myDatabaseServer/theDatabaseName

2.

try {

Class.forName("my.database.driver");

}

catch(Exception ex)

{

System.err.println("Could not load database driver: " + ex);

}

3. Connection conn = DriverManager.getConnection("a.JDBC.URL", "databaseLogin", "databasePassword");

**Question 9**

What is an *iterator* and why are iterators necessary for generic programming?

**Answer :** One of the principle features of Java's generic programming framework is Collections. There are several types of collection (including LinkedLists, ArrayList, TreeSet, and HashSet). In order to deal with all the different types of collection in a generic way, we need a generic way to access all the elements in a collection. An iterator makes this possible. An iterator is an object associated with a collection that makes it possible to traverse the collection (that is, to visit each of the items in the collection in turn). Code written using iterators will work for any type of Collection.

**Question 10**

What is the output of the following code?

String s1 = **new** String("Test");

String s2 = "Test";

**if** (s1==s2)

System.out.println("Same");

**if** (s1.equals(s2))

System.out.println("Equals");

**Answer :** Equals

**Question 11**

When executed, whether or not this code will perform 10 iterations and ends? Why?

**int** x=0;

**int** y=0;

**while**(**true**){

**if** ((++x==10) && (++y==10))

**return**;

}

**Answer :** The optimizer causes the second part of the condition not to be executed when the first part evaluates to false. Consequently, only when x reaches 10, y will become 1. Later, x will overflow and reach 10 again, and so on; until both counters reach 10 and the loop will stop.

**Question 12**

What is the result of attempting to compile and run the following code?

public class Test1{

public static void main(String[] args)

{

Integer int1 = new Integer(10);

Vector vec1 = new Vector();

LinkedList list = new LinkedList();

vec1.add(int1);

list.add(int1);

if(vec1.equals(list))

System.out.println("equal");

else

System.out.println("not equal");

}

}

**Answer :** Will run and print "equal".

**Question 13**

What will happen if you try to compile and run this code.

class Rectangle{

public int area(int length , int width) {

return length \* width;

}

}

class Square extends Rectangle{

}

class Test{

public static void main(String args[]) {

Square r = new Square();

System.out.println(r.area(5 , 4));

}

}

**Answer :** Print 20

**Question 14**

What is the result of attempting to compile and run this ?

interface ITest{

public void setVal();

}

public class Test {

private String a;

void aMethod(){

final String b;

ITest it = new ITest() {

public void setVal(){

a = "Hello";

b = " World";

}

};

it.setVal();

System.out.println(a + b);

}

public static void main(String[] args) {

Test t = new Test();

t.aMethod();

}

}

**Answer :** Code will not compile.

**Question 15**

Whether the following statement is correct? Why?

IOException can be thrown without the need to declare it in the method header.

**Answer :**  Wrong. IOException is a checked exception, and you must either catch it or declare the method as “throws IOException”.

**Question 16**

If you want a servlet to take the same action for both GET and POST request, what should you do?

**Answer :** Simply have doGet call doPost, or vice versa.

**Question 17**

**How to track a user session in Servlets? Write the code that can be used to create session object in the Servlet.**

**Answer :** The interface HttpSession can be used to track the session in the Servlet. Following code can be used to create session object in the Servlet:

**HttpSession session = req.getSession(true);**

**Question 18**

What is a benefit of using JavaBeans to separate business logic from presentation markup within the JSP environment?

**Answer :** It creates a cleaner role separation between the web-production team and the software development team, so that the web-production team can focus on presentation markup, while the software team can focus on building reusable software components for helping to generate dynamic displays.

**Question 19**

How do I include static files within a JSP page?

**Answer :** Static resources should always be included using the JSP include directive. This way, the inclusion is performed just once during the translation phase.   
The following example shows the syntax:   
< % @ include file="copyright.html" % >   
Do note that you should always supply a relative URL for the file attribute. Although you can also include static resources using the action, this is not advisable as the inclusion is then performed for each and every request.

**Question 20**

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.   
The response.sendRedirect method.