**Question 1**

Explain what is a recursive subroutine.

**Question 2**

**What is a binary tree?**

**Question 3**

What is an activation record*?* What role does a stack of activation records play in a computer?

**Question 4**

Explain the following code sample: result = someCondition ? value1 : value2. Give an example.

**Question 5**

You plan to write a program that uses several basic collection interfaces: Set, List, Queue, and Map. You're not sure which implementations will work best, so you decide to use general-purpose implementations. Which implementations are these?

**Question 6**

What are collection pools? What are their advantages?

**Question 7**

What is the output of the following code?

public class JavaTest {

public static void main(String[] args) {

String s1 = new String("Test");

String s2 = "Test";

if (s1==s2)

System.out.println("Same");

if (s1.equals(s2))

System.out.println("Equals");

}

}

**Question 8**

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

**public** **class** JavaTest {

**public** **static** **void** main(String[] args) **throws** InterruptedException {

Thread.*sleep*(2000);

System.*out*.println("Java Quiz");

}

}

**Question 9**

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

public class Test {

public static void main(String[] args){

Float f = new Float(16/0);

System.out.println(f.isNaN());

}

}

**Question 10**

What is the essential difference in functionality between a TreeMap and a HashMap?

**Question 11**

**What will be the output of the following code?**

**class** Base {

**protected** **int** i = 10;

**public** **int** get(){

**return** i;

}

}

**public** **class** Derived **extends** Base{

**protected** **int** i = 20;

**public** **int** get(){

**return** i;

}

**public** **static** **void** main(String argv[])

{

Base b = **new** Derived();

System.out.println(b.i);

System.out.println(b.get());

}

}

**Question 12**

When compiling this code, what will happen? Why?

**class** Parent{

**protected** **void** x(){}

**public** **void** y(){}

}

**public class** Child **extends** Parent{

**public** **void** x(){}

**protected** **void** y(){}

}

**Question 13**

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

public class Test {

public static void main(String[] args){

Integer a = new Integer(4);

Integer b = new Integer(8);

Integer c = new Integer(4);

TreeSet ts = new TreeSet();

ts.add(a);

ts.add(b);

ts.add(c);

System.out.println(ts);

}

}

**Question 14**

What will this program print out ?

class Base{

static int value = 0;

Base(){

addValue();

}

static void addValue(){

value += 10;

}

int getValue(){

return value;

}

}

class Derived extends Base{

Derived(){

addValue();

}

static void addValue(){

value += 20;

}

}

public class Test {

public static void main(String[] args){

Base b = new Derived();

System.out.println(b.getValue());

}

}

**Question 15**

Is there any problem with this exception handler? Will this code compile? What message will be displayed when the code is executed?

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

**try**{

String s = **null**;

s=s.substring(10);

}

**catch**(Exception e){

System.out.println("Caught exception!");

}

**catch**(RuntimeException e){

System.out.println("Caught a runtime exception!");

}

}

**Question 16**

What is the difference between RequestDispatcher and sendRedirect?

**Question 17**

**What do you understand by servlet mapping?**

**Question 18**

Describes the relationship between JSP and servlets.

**Question 19**

What are the implicit objects in JSP technology?

**Question 20**

How can I declare methods within my JSP page?