Q1.Which of the following are interfaces:

1.SortedMap

2.SortedSet

3.ArrayList

4.LinkedHashSet

Q2.Only method in runnable interface:

1.start()

2.run()

3.sleep()

4.join()

Q3.Which of the interface must contain unique element

1.set

2.map

3.arraylist

4.hashmap

Q4.Which class reads and write data to or from file in form of bytes

1.Stream Class

2.ReaderWriter Class

3.Scanner class

4.none

Q5.Which are the correct methods in Assert class:

1.fail(String)

2.assertTrue(Boolean)

3.assertNull([msg],Object)

4.all of the above

Q6.Serializable is a\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.concrete class

2.marker interface

3.functional interface

4.final class

Q7. **public** **class** sampleclass

{

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

Map<String,String> map=**new** TreeMap();

map.put("c","2");

map.put("a","3");

map.put("b","1");

System.***out***.println(map);

}

1. {a=3, b=1, c=2}
2. {a=1, b=2, c=3}
3. {b=1, c=2, a=3 }
4. (c=2,a=3,b=1}

Q8.Which keyword is used to make an property not available for serializable

1.transient

2.protected

3.private

4.undefined

Q9.The default capacity of arraylist is

1.12

2.6

3.7

4.10

Q10.How to test exception in junit testing

1.@Test(expected=ArithmeticException.class)

2. @Test(expected=”ArithmeticException.class”)

3. .@Test(expected=ArithmeticException)

4. [.@Test(exception=AruthmeticException.class)](mailto:.@Test(exception=AruthmeticException.class))

Q11.Which of the following lamda expression is true:

1. (num) -> num+10
2. (int num1,int num2)->{int min=num1>num2?num2:num1;

return min}

3. () ->”hello world”

4. all

Q12.

**public** **class** sampleclass

{

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

Map<String,String> map=**new** HashMap();

map.put(**new** String("a"),"hello");

map.put("a","world");

System.***out***.println(map);

}

}

1.{a=hello,a=world}

2.{a=hello}

3.{a=world}

4.exception

Q13.

@FunctionalInterface

**interface** StringConcat

{

**public** String concat(String a,String b);

}

**public** **class** sampleclass

{

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

StringConcat s=(st1,st2)-> st1+st2;

System.***out***.println(s.concat("hello","hi"));

}

}

1.hellohi

2.null

3.hello hi

Q14.

**public** **class** sampleclass

{

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

List<String> list =**new** ArrayList<>();

list.add("Hrishu");

list.add("Anmol");

list.add("Adi");

list.forEach(

(names)->System.***out***.print(names));

}

}

1.Hrishu Anmol Adi

2.null null null

3.compilation error

4.exception

Q15. Which of the following are the operations of stream:

1.Filter

2.Map

3.Reduce

4.All

Q16.Which of the following are the characteristics of stream:

1.designed for lamdas

2.do not support indexed access

3.can easily be output as array and list

4.all

**Q17. Find the output of the following code snipet:**

**class** Base

{

**public** **static** **void** show()

{

System.***out***.println("base");

}

}

**class** Derived **extends** Base

{

**public** **static** **void** show()

{

System.***out***.println("derived");

}

}

**public** **class** sampleclass

{

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

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

b.*show*();

}

}

1.derived

2.base

3.compilation error

4.base derived

Q18.which of the following are true about abstract:

1.abstract class may or may not contain abstract method

2.abstract class can have static and non static member variables

3.has constructor

4.can extend more than one abstract class

Q19.What is the exception raised in this code

Object i=Integer.valueOf(42);

String s=(String)i;

1.ClassCastException

2.TypeCastException

3.IllegalArgumentException

4.NullPointerException

Q20.Which of the following are checked exception

1.SQLException

2.ClassNotFoundException

3.InterruptedException

4.NullPointerException

Q21.Can implement try without a catch block ?

1.true

2.false

Q22. which of the following are true about interface:

1.it can have any access modifiers

2.it can only static or final variables

3.default and static method can have implemented body

4.can have constructor

Q23.which of the following is the correct way to define a 2d array:

1.int a[][]=new int[4][5]

2. int a[][]=new int[4][]

3. int a[][]=new int[][]{{3,5},{4,7},{9,5}}

4. int a[][]=new int[][5]

Q24.which of the following are the ways to close database or file connection

1.try with resources

2.finally

3.catch with resources

4.none

Q25.Which are the correct methods in Assert class:

1.fail(String)

2.assertTrue(Boolean)

3.assertNull([msg],Object)

4.all of the above

Q26.

**public** **class** sampleclass

{

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

**int** t;

System.***out***.println(t);

}

}

1.0

2.garbage value

3.compilation error

4.exception occur

Q27.

**public** **class** sampleclass

{

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

Map<String,String> map=**new** TreeMap();

map.put("c","2");

map.put("a","3");

map.put("b","1");

System.***out***.println(map);

}

1. {a=3, b=1, c=2}
2. {a=1, b=2, c=3}
3. {b=1, c=2, a=3 }
4. (c=2,a=3,b=1}

Q28.Which is true about enum:

1. a semicolon after enum is optional
2. enum can be declared only public and default access modifiers
3. both 1 and 2
4. none

Q29.Which of the interface must contain unique element

1.set

2.map

3.arraylist

4.hashmap

Q30.How to avoid duplicate object in treeset:

1.overriding hashcode() and equals() in that particular class

2. overriding hashcode() only

3.overriding equals() only

4.we cant avoid duplication

Q31.

**interface** AddSub

{

**public** **int** add(**int** a,**int** b);

**public** **int** sub(**int** a,**int** b);

}

**public** **class** sampleclass

{

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

AddSub m=(st1,st2)-> st1+st2;

System.***out***.print(m.add(5,5));

AddSub m=(st1,st2)-> st1-st2;

System.***out***.print(m.sub(5,4));

}

}

1. compilation error

2. 10 1

3. exception

4. 1 1

Q32. Which of the collection has the ability to grow dynamically

1.Array

2.Arrays

3.ArrayList

4.none

Q33. Who compiles bytecode to platform specific executable code:

1.JDK

2.JRE

3.JIT

4.None

Q34.Which operation returns unique element:

1.filter(Predicate)

2.distinct

3.limit(n)

4.none

Q35. Which annotation makes an test unavailable for testing

1.@Ignore

2.@Disable

3.@Unable

4.@NotTest