Akash

**Q1. Find the output of the following code snipet:**

**class** Base

{

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

{

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

}

}

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

{

**public** **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

Q2. Which of the following is true:

**class** demo

{

**int** a=1;

**void** fun()

{

innerclass obj=**new** innerclass();

obj.display();

}

**class** innerclass

{

**int** b=2;

**void** display()

{

System.***out***.print(a);

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

}

}

**void** get()

{

System.***out***.print(a);

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

}

}

**public** **class** sampleclass

{

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

demo ob=**new** demo();

ob.fun();

ob.get();

}

}

1. 12

12

2.Compilation error as get() cannnot use variable b of innerclass

3.Compilation error as display() cannnot use variable a of outerclass

4.we cannot create object of innerclass inside outer class

Q3.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]

Q4.

Class A

{

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

String ob=**new** String("hello");

String ob2="hello";

**if**(ob.hashCode()==ob2.hashCode())

{

System.***out***.println("hashcode equal");

}

**if**(ob==ob2)

{

System.***out***.println("memory address equal");

}

**if**(ob.equals(ob2))

{

System.***out***.println("value equal");

}

}

}

1. hashcode equal

memory address equal

value equal

1. hashcode equal

value equal

1. value equal

4. memory address equal

value equal

Q5.

**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

Q6. 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

Q7.Whcih of the following annotations are present in java:

1.@Override

[2.@SuppressWarnings](mailto:2.@SuppressWarnings)

3.@Description

[4.@Deprecated](mailto:4.@Deprecated)

Q8.Which are the methods of object class:

1.class getClass()

2.void finalize()

3.Object clone()

4.void run()

Q9.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

Q10. Who compiles bytecode to platform specific executable code:

1.JDK

2.JRE

3.JIT

4.None

Q11.

**public** **class** sampleclass

{

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

{

**int** a,b,c;

**try**

{

**throw** **new** ClassNotFoundException();

**throw** **new** ArrayIndexOutOfBoundsException();

System.***out***.println("enter two integers");

Scanner sc= **new** Scanner(System.***in***);

b=sc.nextInt();

a=sc.nextInt();

c=a/b;

System.***out***.println("result is"+c);

**throw** **new** NullPointerException();

}

**catch**(ArithmeticException | NullPointerException e)

{

System.***out***.println("Arithmetic or Nullpointer ");

}

**catch**( ClassNotFoundException| ArrayIndexOutOfBoundsException e)

{

System.***out***.println("Classnotfound or Arrayindexoutofbound");

}

**catch**(Exception e)

{

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

}

**finally**

{

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

}

}

}

1. Classnotfound or Arrayindexoutofbound

Finally

1. Arithmetic or Nullpointer

Finally

1. compile time error

Arithmetic or Nullpointer

Classnotfound or Arrayindexoutofbound

Finally

Q12.

**public** **class** sampleclass

{

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

{

**int** a,b,c;

**try**

{

**try**{

System.***out***.println("enter two integers");

Scanner sc= **new** Scanner(System.***in***);

a=sc.nextInt();

c=a/0;

System.***out***.println("result is"+c);}

**catch**(ArithmeticException e)

{

System.***out***.println("arithmeticexception inner try");

}**finally**

{

System.***out***.println("inner finally");

}

}

**catch**(ArithmeticException e)

{

System.***out***.println("arithmeticexception outer try");

}

**catch**(Exception e)

{

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

}

**finally**

{

System.***out***.println("outer Finally");

}

}

}

1. arithmeticexception inner try

inner finally

outer Finally

1. arithmeticexception inner try

inner finally

arithmeticexception outer try

outer Finally

1. Compile time error

4. arithmeticexception inner try

inner finally

Q13.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

Q14.Which of the following is true

1.each try block can have only one finally

2.each try block can have multiple catch

3.both

4.none

Q15.

How can we handle uncheckedException?

1.try-catch block

2.throws keyword

3.both 1 and 2

4.none

Q16.Which exception occur when we try to start a thread which is already started

1.InterruptedException

2.IllegalSateException

3.IllegalArgumentException

4.ClassCastException

Q17.Which are the ways to make a thread into a blocked state

1.by calling sleep()

2.suspend()

3.wait()

4.notifyAll()

Q18.Serializable is a\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.concrete class

2.marker interface

3.functional interface

4.final class

Q19.Which keyword is used to make an property not available for serializable

1.transient

2.protected

3.private

4.undefined

Q20.which of the following are correct annotations in junit

1.@test

2.@Before

3.@Ignore

4.all

Q21.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))

Q22.

**public** **class** sampleclass

{

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

Set hashSet=**new** HashSet();

hashSet.add("1");

hashSet.add(1);

hashSet.add(**null**);

hashSet.add(**null**);

hashSet.add("null");

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

}

}

1.null,1

2.null,1,1,null

3.null,null,1,1,null

4.compile time error

Q23.

**public** **class** sampleclass

{

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

List list=**new** LinkedList();

list.add(**new** Integer(5));

list.add(6);

list.add(**new** String("7"));

list.add("8");

list.add(**null**);

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

}

}

1.compilation error

2.ClassCastException

3.5 6 7 8 null

4.NumberFormatException

Q24.

**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

Q25.Which of the interface must contain unique element

1.set

2.map

3.arraylist

4.hashmap

Q26.What implementation of iterator can iterate in both direction

1.ListIterator

2.Iterator

3.MapIterator

4.SetIterator

Q27.HashSet internally uses :

1.HashMap

2.Set

3.List

4.Collection

Q28.Which of the following are interfaces:

1.SortedMap

2.SortedSet

3.ArrayList

4.LinkedHashSet

Q29.Which of the following is true

1.lamda expression is an anonymous block of code that encapsulate an expression or a block of statement and returns a result

2.functionalinterface has single abstract method only

3.return type of λE is functionalinterface

4.all

Q30.

**public** **class** sampleclass

{

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

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

list.add("akash");

list.add("aindrila");

list.add("arijeet");

list.forEach(

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

}

}

1.akash aindrila arijeet

2.null null null

3.compilation error

4.exception

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.

**class** Student

{

**private** String name="Student Object";

@Override

**public** String toString() {

**return** "Student [name=" + name + "]";

}

}

**class** Sender<T>

{

**private** T message;

**public** **void** setMsg(T message)

{

**this**.message=message;

}

**public** **void** printmsg()

{

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

}

}

**public** **class** sampleclass

{

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

Sender<String> obStr=**new** Sender();

obStr.setMsg("Hello");

obStr.printmsg();

Sender<Student> obStudent=**new** Sender();

obStudent.setMsg(**new** Student());

obStudent.printmsg();

}

}

1. Hello

Student [name=Student Object]

2.null

null

3.compilation error

4.exception

Q33.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

Q34. A stream implementation may throw \_\_\_\_\_\_\_\_\_\_\_\_\_ if it detects that the stream is being reduced

1. IllegalStateException
2. ClassCastException
3. IllegalArgumentException
4. IOException

Q35. How many ways we can synchronized thread

1. synchronized method
2. synchronized block
3. synchronized static method
4. all