**Java SE**

Q. No. 1

**Question:**

What makes Java SE a popular choice among solution developers?

Answer Choices

A: Automatic Memory Management

B: Architectural neutral

C: Secure

D: All of above.

Q. No. 2

**Question:**

Which of the following is NOT true while comparing C++ with Java?

Answer Choices

A: Java class type of references are equivalent to C++ object pointers.

B: Java does not have virtual keyword like C++ , but supports run time polymorphism.

C: Java does not support multiple inheritance , in C++ it is supported.

D: Java generics support is exactly like templating in C++ , so java also suffers from template code bloat.

Q. No. 3

**Question:**

Which of the following is true regarding correct overriding form of methods?

Answer Choices

A: Overriding form of the method can return, super type of the overridden form.

B: Overriding form of the method can assign weaker access specifier than overridden form.

C: Overriding form of the method can add in its throws clause , any new unchecked exception .

D: Overriding form of the method can add in its throws clause , any new checked exception .

Q. No. 4

**Question:**

Given:

11. public static void main(String[] args) {

12. Object obj = new int[] { 1, 2, 3 };

13. int[] someArray = (int[])obj;

14. for (int i : someArray) System.out.print(i + " ");

15. }

What is the result?

Answer Choices

A: 123

B: Compilation error at line 12

C: Compilation error at line 13

D: Run time error of ClassCastException

Q. No. 5

**Question:**

Given:

11. public static void main(String[] args) {

12. byte b1=0x15;

13. byte b2=0x30;

14. byte b3 = b1+b2;

15. byte b4;

16. b4 += b1;

17. b3 += b1;

18. float f1=12.34;

}

What is the result?

Answer Choices

A: Run time error.

B: Compiler error at line 14,16,18

C: Compiler error at line 14,15,16,18

D: Compiler error at line 15,18

Q. No. 6

**Question:**

Given :

public class TestOverLoad { public static void main(String[] args) {

test(null);

}

public static void test(String e)

{

System.out.println("in string");

}

public static void test(Integer e)

{

System.out.println("in Integer");

} }

Answer Choices

A: Compiler error

B: Run time error

C: Displays output --- in string

D: Displays output --- in Integer

Q. No. 7

**Question:**

A protected instance method in the superclass can be made \_\_\_\_\_\_\_ scope , but not \_\_\_\_\_\_\_\_scope , in the subclass.

Answer Choices

A: public,default

B: protected,public

C: default,public

D: private,default

Q. No. 8

**Question:**

Can we declare variable inside a method as final variable and Can an abstract class may be final?

Answer Choices

A: Yes,Yes

B: Yes,No

C: No,Yes

D:No,No

Q. No. 9

**Question:**

Given : class Base

{

public Object getValue(){ return new Object(); } //1

}

class Base2 extends Base

{

public String getValue(){ return "hello"; } //2

}

public class TestClass

{

public static void main(String[] args)

{

Base b = new Base2();

System.out.println(b.getValue()); //3

}

}

What will be output?

Answer Choices

A: Compiler error at line 2

B: Compiler error at line 3

C: Prints hello

D: Prints hash code of object

Q. No. 10

**Question:**

State which of the following statements are true .

a)   An abstract class must have at least one abstract method

b)   An abstract class can have 0 - n methods abstract but even if one method is declared as abstract the class must be declared abstract.

c)   An abstract method can have a method body.

d)   An abstract class can be instantiated.

Answer Choices

A: a,b B: b

C: b,c D: a,b,c

Q. No. 11

**Question:**

What will be the result of compiling and running the following code?

public class Sample implements IInt

{

public static void main(String[] args)

{

Sample s = new Sample(); //1

int j = s.thevalue; //2

int k = IInt.thevalue; //3

int l = thevalue; //4

}

}

interface IInt

{

int thevalue = 0;

}

What will be output?

Answer Choices

A: It will give an error at compile time at line //1.

B: It will give an error at compile time at line //2

C: It will give an error at compile time at line //4

D: No compiler error & no runtime errors.

Q. No. 12

**Question:**

class Test

{

public static void main(String[] ss)

{

try {

testMe(100,0); // 1

} catch (InterruptedException e)

{}

System.out.println("main over..");

}

static void testMe(int i1,int i2) throws Exception

{

System.out.println("Result "+(i1/i2)); // 2

Thread.sleep(100); // 3

System.out.println("Method over..");

}

}

What will be output?

Answer Choices

A: Run time error handled by JVM’s default handler

B: Prints main over

C: Compiler error at line 2 & 3

D: Compiler error at line 1

Q. No. 13

**Question:**

import java.util.Scanner;

public class Test2 {

public static void main(String[] args) {

try (Scanner sc=new Scanner("hello hi 1234"))

{

System.out.println(sc.nextInt()); //1

} catch (Exception e) {

System.out.println(e.getMessage());

}

sc.close(); //2

}

}

Answer Choices

A: Prints null

B: Compiler error at 1

C: Compiler error at 2

D: Prints InputTypeMismatchException

Q. No. 14

**Question:**

import java.util.\*;

public class Test3 {

public static void main(String[] args) {

List<Integer> l1=Arrays.asList(12, 23, 11, 34, 12,45, 34);

l1.add(19); //1

HashSet<Integer> hs = new HashSet<>(l1);//2

hs.add(123);//3

System.out.println("HS "+hs);//4

}

}

Answer Choices

A: Compiler error at 1, 2

B: Compiler error at 3

C: Run time error : UnsupportedOperationException

D: Prints contents of HashSet at line 4, by removing duplicates.

Q. No. 15

**Question:**

Given :

import java.util.\*;

public class Test4 {

public static void main(String[] args) {

List<A> l1=new LinkedList<>(); //1

l1.add(new B()); //2

HashSet<B> hs = new HashSet<>(l1); //3

hs.add(new B()); //4

System.out.println("HS "+hs);

}

}

class A {}

class B extends A{}

What will be output ?

Answer Choices

A: Compiler error at line 1

B: Complier error at line 2

C: Complier error at line 3

D: Complier error at line 3 & 4

Q. No. 16

Question:

import java.util.\*;

public class Test5 {

public static void main(String[] args) {

List<Integer> l1=Arrays.asList(12, 23, 11, 34, 12,45, 34);

Iterator<Integer> itr=l1.iterator();//1

System.out.println("Removing 1st element");

itr.remove();//2

System.out.println("List upon removal "+l1);//3

}

}

Answer Choices

A: At line no 3 prints list contents after removing 1st element

B: Compiler error at line 2

C: At line no 3 prints list contents without removing 1st element

D: Run time error at line 2

Q. No. 17

Question:

import java.util.HashMap;

public class Test6 {

public static void main(String[] args) {

String[] names={"abc","abc123","hello","123","xyz"};

HashMap<Integer, String> hm=new HashMap<>();

for(String s : names)

hm.put(s.length(), s);

System.out.println(hm); //1

}

}

Answer Choices

A: Prints null from line 1

B: Null Pointer Exception at line 1

C: Prints 3 entries from from line 1

D: Prints 5 entries from from line 1

Q. No. 18

Question:

Given

import java.util.\*;

public class Test6 {

public static void main(String[] args) {

String[] names={"abc1","abc123","hello","123","xyz"};

HashMap<Integer, String> hm=new HashMap<>();

for(String s : names)

hm.put(s.length(), s);

Collection<String> c=hm.values();

Iterator<String> itr=c.iterator();//1

hm.put(4, "abcd");//2

while (itr.hasNext())

System.out.print(itr.next() +" ");//3

itr=c.iterator();

hm.put(1, "a");

while (itr.hasNext())

System.out.print(itr.next() +" ");//4

}

}

Answer Choices

A: Compiler Error

B: Run time error at 3

C: Run time error at 4

D: Prints 7 strings from line 4

Q. No. 19

Question:

Given the following classes and declarations, which of these statements about //1 and //2 are true?

class A

{

private int i = 10;

public void f(){}

public void g(){}

}

class B extends A

{

public int i = 20;

public void g(){}

}

public class C

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

A a = new A();//1

A b = new B();//2

} }

Answer Choices

A: System.out.println(b.i); will print 10.

B: The statement b.f( ); will give compile time error..

C: System.out.println(b.i); will print 20

D: None of the above statements are correct.

Q. No. 20

Question:

Given:

1. enum Animals {

2. DOG("woof"), CAT("meow"), FISH("burble");

3. String sound;

4. Animals(String s) { sound = s; }

5. }

6. class TestEnum {

7. static Animals a;

8. public static void main(String [] args) {

9. System.out.println(a.DOG.sound + " " + a.FISH.sound);

10. }

11. }

What is the result?

Answer Choices

A: Multiple compilation errors

B: woof burble

C: Compilation fails due to an error on line 3

D: Compilation fails due to an error on line 9

Q. No. 21

Question:

Given two files:

1. package pkgA;

2. public class Foo {

3. int a = 5;

4. protected int b = 6;

5. public int c = 7;

6. }

3. package pkgB;

4. import pkgA.\*;

5. public class Baz {

6. public static void main(String[] args) {

7. Foo f = new Foo();

8. System.out.print(" " + f.a);

9. System.out.print(" " + f.b);

10. System.out.print(" " + f.c);

11. }

12. }

What is the result?

Answer Choices

A: Compilation error at line 8 & 9

B: Compilation error at line 8

C: Compilation error at line 9

D: prints 5 6 7

Q. No. 22

Question:

Given:

class Clidder {

private final void flipper() { System.out.println("Clidder"); }

}

public class Clidlet extends Clidder {

public final void flipper() { System.out.println("Clidlet"); }

public static void main(String [] args) {

new Clidlet().flipper();

} }

What is the result?

Answer Choices

A: compilation error

B: Prints Clidder

C: Prints Clidlet

D: Prints Clidder

Cliddlet

Q. No. 23

Question:

What are all the methods used for Inter Thread communication and what is the class in which these methods are defined?

Answer Choices

A: wait(),notify() & notifyAll() belonging to Thread class.

B: wait(),notify() & notifyAll() belonging to Object class.

C: wait(),start() & run() belonging to Thread class.

D: interrupt(),join() & wakeup() belonging to Object class.

Q. No. 24

Question:

Given following 2 classes .

import java.util.\*;

class NameList {

List<String> names =new LinkedList<>();

public void add(String name) {

names.add(name);

}

public String removeFirst() {

if (names.size() > 0) {

try {

Thread.sleep(10);

} catch (InterruptedException e) {System.out.println(“error”);}

return names.remove(0);

} else

return null;

}

}

class TestSafeCollections {

public static void main(String[] args) {

final NameList nl = new NameList();

nl.add("abc");

class NameDropper extends Thread {

public void run() {

synchronized (nl) {

String name = nl.removeFirst();

System.out.println(name);

}

}

}

Thread t1 = new NameDropper();

Thread t2 = new NameDropper();

t1.start();

t2.start();

}

}

What will be the result?

Answer Choices

A: Prints abc

null

B: Prints abc & then raises IndexOutOfBoundsException

C: Prints error

D: Raises IllegalStateException

Q. No. 25

Question:

public class TestThreads {

public static void main(String[] args) {

Runnable r1=new Runnable() {

@Override

public void run() {

System.out.println("Invoked by +Thread.currentThread().getName());

}

};

Thread t1=new Thread(r1, "thread-1");

t1.run();

}

}

What will be the result ?

Answer Choices

A: Compiler error

B: Run time error

C: Prints “Invoked by thread-1”

D: None of above

Q. No. 26

Question:

Given :

class Utils {

synchronized void test()

{

System.out.println("entered test "+Thread.currentThread().getName());

try {

Thread.sleep(1000);

} catch (InterruptedException e) {}

System.out.println("exited test "+Thread.currentThread().getName());

}

synchronized void testMe()

{

System.out.println("entered testMe "+Thread.currentThread().getName());

try {

Thread.sleep(1000);

} catch (InterruptedException e) {}

System.out.println("exited test "+Thread.currentThread().getName());

}

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

final Utils u1 = new Utils();

Runnable r1 = new Runnable() {

@Override

public void run() { while (true)

u1.test(); }

};

Runnable r2 = new Runnable() {

@Override

public void run() {

while (true)

u1.testMe();

}

};

Thread t1 = new Thread(r1, "t1");

Thread t2 = new Thread(r2, "t2");

t1.start();

t2.start();

}

}

Answer Choices

A: Possible output

entered test t1

entered testMe t2

exited test t1

exited testMe t2

B: Possible output

entered testMe t2

entered test t1

exited testMe t2

exited test t1

C: Possible output

entered test t1

entered testMe t2

exited testMe t2

exited test t1

D: None of above

Q. No. 27

Question:

Given

import javax.swing.\*;

public class SwingFrame extends JFrame

{

public static void main(String[] args) {

JFrame f1=new SwingFrame();

}

public SwingFrame() {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

System.out.println(“Invoked by “+Thread.currentThread().getName());

}

});

}

}

Answer Choices

A: Run time error

B: No error , prints blank

C: Prints “Invoked by AWT-EventQueue-0”

D: Prints “Invoked by main”

Q. No. 28

Question:

What are the features of PrintWriter?

Answer Choices

A: Supplies buffering & automatic flushing of data.

B: Can wrap any binary as well as char stream

C: Supports print,println,printf API methods

D: All of above.

Q. No. 29

Question:

Which of the following statements is NOT true?

Answer Choices

A: If super class is serializable , sub-class is automatically serializable.

B: During de-serialization, default constructor is always called.

C: During de-serialization, binary stream containing state of the object & its corresponding Class object(Class<T>) are both essential

D: During serialization , transient & static data members are skipped from serialization.

Q. No. 30

Question:

Which of the following statements is true?

Answer Choices

A: Non static nested classes can not contain any static member.

B: Non static nested classes can not access outer class’s private members directly.

C: Static nested class can directly access non static member of outer class.

D: Method local inner class can access non final method local variable .

Q. No. 31

Question:

Given:

3. class SubException extends Exception { }

4. class SubSubException extends SubException { }

5.

6. public class CC { void doStuff() throws SubException { } }

7.

8. class CC2 extends CC { void doStuff() throws SubSubException { } }

9.

10. class CC3 extends CC { void doStuff() throws Exception { } }

11.

12. class CC4 extends CC { void doStuff(int x) throws Exception { } }

13.

14. class CC5 extends CC { void doStuff() { } }

What is the result?

Answer Choices

A: Compilation fails due to an error on line 8

B: Compilation fails due to an error on line 10

C: Compilation fails due to an error on line 12

D: No compiler error

Q. No. 32

Question:

Given:

3. public class Ebb {

4. static int x = 7;

5. public static void main(String[] args) {

6. String s = "";

7. for(int y = 0; y < 3; y++) {

8. x++;

9. switch(x) {

10. case 8: s += "8 ";

11. case 9: s += "9 ";

12. case 10: { s+= "10 "; break; }

13. default: s += "d ";

14. case 13: s+= "13 ";

15. }

16. }

17. System.out.println(s);

18. }

19. static { x++; }

20. }

What is the result?

Answer Choices

A: 8 9 10 d

B: 9 10 10 d

C: 9 10 10 d 13

D: 8 9 10 10 d 13

Q. No. 33

Question:

What is essential to make a deep clone copy of any object of user defined type eg Emp?

Answer Choices

A: The class must implement Cloneable interface

B: Must override clone() method properly

C: A & B both

D: Neither A nor B is essential

Q. No. 34

Question:

Given :

public class TestStrings

{

public static void main(String[] args)

{

String one = "someString";

String two = "someString";

String three=new String(one);

System.out.print(one.equals(two));

System.out.print(one == two);

System.out.print(one.equals(three));

System.out.print(one == three);

}

}

Answer Choices

A: true true true true

B: true false true false

C: true true false true

D: true true true false

Q. No. 35

Question:

Given

public class ImmutableStrings

{

public static void main(String[] args)

{

String one = "someString";

String two = new String("someString");

String three=”someString”;

one = two = three=null;

System.out.println(“testing ”);//1

System.out.println(“testing again ”);//2

}

}

At line no 1 , how many objects are marked for garbage collection?

Answer Choices

A: 0 B: 1

C: 2 D: 3

Q. No. 36

Question:

Consider following business requirement. Multiple bank branches need to upload suitable collection of bank accounts to a centralized server. Upon receiving collection server sends text acknowledgement to the clients & then saves this received collection to binary file . Assume that clients & server is developed using TCP sockets. Suggest suitable networking data streams that , a client application has to open for this reuirement.

Answer Choices

A: ObjectInputStream, BufferedReader

B: Scanner, PrintWriter

C: FileOutputStream, BufferedReader

D: ObjectOutputStream, BufferedReader

Q. No. 37

Question:

What is mandatory contract between overriding of hashCode & equals , for correct working of HashSet ?

Answer Choices

A: No such contract exists

B: If 2 objects are equal via equals method, then they must produce same integer hash codes.

C: If 2 objects are un equal via equals method, then they must produce distinct integer hash codes.

D: B & C both are mandatory

Q. No. 38

Question:

Which of following is NOT true?

Answer Choices

A: Final classes can not be extended.

B: Since abstract classes can’t be instantiated, they can’t contain any constructor.

C: Abstract class may contain complete concrete functionality.

D: Final references can not be re assigned.

Q. No. 39

Question:

Given :

import java.util.\*;

public class Test8 {

public static void main(String[] args) {

List<Integer> l1=Arrays.asList(12,1,234,145,100,45,87);

Collections.sort(l1, new Comparator<Integer>() {

@Override

public int compare(Integer o1, Integer o2) {

return o2-o1;

}});

System.out.println("List : "+l1);

}

}

What is the result ?

Answer Choices

A: Compiler error.

B: ClassCast Exception

C: List contents printed as per ascending order.

D: List contents printed as per descending order.

Q. No. 40

Question:

What will happen if a class extends Thread , does not override run() method & its object is created & invoked start() on the same object?

Answer Choices

A: Compiler Error

B: Run time error

C: Thread is permanently blocked

D: None of above

**Java SE**

Q. No. 1

**Question:**

What makes Java SE a popular choice among solution developers?

Answer Choices

A: Automatic Memory Management

B: Architectural neutral

C: Secure

D: All of above.

Q. No. 2

**Question:**

Which of the following is NOT true while comparing C++ with Java?

Answer Choices

A: Java class type of references are equivalent to C++ object pointers.

B: Java does not have virtual keyword like C++ , but supports run time polymorphism.

C: Java does not support multiple inheritance , in C++ it is supported.

D: Java generics support is exactly like templating in C++ , so java also suffers from template code bloat.

Q. No. 3

**Question:**

Which of the following is true regarding correct overriding form of methods?

Answer Choices

A: Overriding form of the method can return, super type of the overridden form.

B: Overriding form of the method can assign weaker access specifier than overridden form.

C: Overriding form of the method can add in its throws clause , any new unchecked exception .

D: Overriding form of the method can add in its throws clause , any new checked exception .

Q. No. 4

**Question:**

Given:

11. public static void main(String[] args) {

12. Object obj = new int[] { 1, 2, 3 };

13. int[] someArray = (int[])obj;

14. for (int i : someArray) System.out.print(i + " ");

15. }

What is the result?

Answer Choices

A: 123

B: Compilation error at line 12

C: Compilation error at line 13

D: Run time error of ClassCastException

Q. No. 5

**Question:**

Given:

11. public static void main(String[] args) {

12. byte b1=0x15;

13. byte b2=0x30;

14. byte b3 = b1+b2;

15. byte b4;

16. b4 += b1;

17. b3 += b1;

18. float f1=12.34;

}

What is the result?

Answer Choices

A: Run time error.

B: Compiler error at line 14,16,18

C: Compiler error at line 14,15,16,18

D: Compiler error at line 15,18

Q. No. 6

**Question:**

Given :

public class TestOverLoad { public static void main(String[] args) {

test(null);

}

public static void test(String e)

{

System.out.println("in string");

}

public static void test(Integer e)

{

System.out.println("in Integer");

} }

Answer Choices

A: Compiler error

B: Run time error

C: Displays output --- in string

D: Displays output --- in Integer

Q. No. 7

**Question:**

A protected instance method in the superclass can be made \_\_\_\_\_\_\_ scope , but not \_\_\_\_\_\_\_\_scope , in the subclass.

Answer Choices

A: public,default

B: protected,public

C: default,public

D: private,default

Q. No. 8

**Question:**

Can we declare variable inside a method as final variable and Can an abstract class may be final?

Answer Choices

A: Yes,Yes

B: Yes,No

C: No,Yes

D:No,No

Q. No. 9

**Question:**

Given : class Base

{

public Object getValue(){ return new Object(); } //1

}

class Base2 extends Base

{

public String getValue(){ return "hello"; } //2

}

public class TestClass

{

public static void main(String[] args)

{

Base b = new Base2();

System.out.println(b.getValue()); //3

}

}

What will be output?

Answer Choices

A: Compiler error at line 2

B: Compiler error at line 3

C: Prints hello

D: Prints hash code of object

Q. No. 10

**Question:**

State which of the following statements are true .

a)   An abstract class must have at least one abstract method

b)   An abstract class can have 0 - n methods abstract but even if one method is declared as abstract the class must be declared abstract.

c)   An abstract method can have a method body.

d)   An abstract class can be instantiated.

Answer Choices

A: a,b B: b

C: b,c D: a,b,c

Q. No. 11

**Question:**

What will be the result of compiling and running the following code?

public class Sample implements IInt

{

public static void main(String[] args)

{

Sample s = new Sample(); //1

int j = s.thevalue; //2

int k = IInt.thevalue; //3

int l = thevalue; //4

}

}

interface IInt

{

int thevalue = 0;

}

What will be output?

Answer Choices

A: It will give an error at compile time at line //1.

B: It will give an error at compile time at line //2

C: It will give an error at compile time at line //4

D: No compiler error & no runtime errors.

Q. No. 12

**Question:**

class Test

{

public static void main(String[] ss)

{

try {

testMe(100,0); // 1

} catch (InterruptedException e)

{}

System.out.println("main over..");

}

static void testMe(int i1,int i2) throws Exception

{

System.out.println("Result "+(i1/i2)); // 2

Thread.sleep(100); // 3

System.out.println("Method over..");

}

}

What will be output?

Answer Choices

A: Run time error handled by JVM’s default handler

B: Prints main over

C: Compiler error at line 2 & 3

D: Compiler error at line 1

Q. No. 13

**Question:**

import java.util.Scanner;

public class Test2 {

public static void main(String[] args) {

try (Scanner sc=new Scanner("hello hi 1234"))

{

System.out.println(sc.nextInt()); //1

} catch (Exception e) {

System.out.println(e.getMessage());

}

sc.close(); //2

}

}

Answer Choices

A: Prints null

B: Compiler error at 1

C: Compiler error at 2

D: Prints InputTypeMismatchException

Q. No. 14

**Question:**

import java.util.\*;

public class Test3 {

public static void main(String[] args) {

List<Integer> l1=Arrays.asList(12, 23, 11, 34, 12,45, 34);

l1.add(19); //1

HashSet<Integer> hs = new HashSet<>(l1);//2

hs.add(123);//3

System.out.println("HS "+hs);//4

}

}

Answer Choices

A: Compiler error at 1, 2

B: Compiler error at 3

C: Run time error : UnsupportedOperationException

D: Prints contents of HashSet at line 4, by removing duplicates.

Q. No. 15

**Question:**

Given :

import java.util.\*;

public class Test4 {

public static void main(String[] args) {

List<A> l1=new LinkedList<>(); //1

l1.add(new B()); //2

HashSet<B> hs = new HashSet<>(l1); //3

hs.add(new B()); //4

System.out.println("HS "+hs);

}

}

class A {}

class B extends A{}

What will be output ?

Answer Choices

A: Compiler error at line 1

B: Complier error at line 2

C: Complier error at line 3

D: Complier error at line 3 & 4

Q. No. 16

Question:

import java.util.\*;

public class Test5 {

public static void main(String[] args) {

List<Integer> l1=Arrays.asList(12, 23, 11, 34, 12,45, 34);

Iterator<Integer> itr=l1.iterator();//1

System.out.println("Removing 1st element");

itr.remove();//2

System.out.println("List upon removal "+l1);//3

}

}

Answer Choices

A: At line no 3 prints list contents after removing 1st element

B: Compiler error at line 2

C: At line no 3 prints list contents without removing 1st element

D: Run time error at line 2

Q. No. 17

Question:

import java.util.HashMap;

public class Test6 {

public static void main(String[] args) {

String[] names={"abc","abc123","hello","123","xyz"};

HashMap<Integer, String> hm=new HashMap<>();

for(String s : names)

hm.put(s.length(), s);

System.out.println(hm); //1

}

}

Answer Choices

A: Prints null from line 1

B: Null Pointer Exception at line 1

C: Prints 3 entries from from line 1

D: Prints 5 entries from from line 1

Q. No. 18

Question:

Given

import java.util.\*;

public class Test6 {

public static void main(String[] args) {

String[] names={"abc1","abc123","hello","123","xyz"};

HashMap<Integer, String> hm=new HashMap<>();

for(String s : names)

hm.put(s.length(), s);

Collection<String> c=hm.values();

Iterator<String> itr=c.iterator();//1

hm.put(4, "abcd");//2

while (itr.hasNext())

System.out.print(itr.next() +" ");//3

itr=c.iterator();

hm.put(1, "a");

while (itr.hasNext())

System.out.print(itr.next() +" ");//4

}

}

Answer Choices

A: Compiler Error

B: Run time error at 3

C: Run time error at 4

D: Prints 7 strings from line 4

Q. No. 19

Question:

Given the following classes and declarations, which of these statements about //1 and //2 are true?

class A

{

private int i = 10;

public void f(){}

public void g(){}

}

class B extends A

{

public int i = 20;

public void g(){}

}

public class C

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

A a = new A();//1

A b = new B();//2

} }

Answer Choices

A: System.out.println(b.i); will print 10.

B: The statement b.f( ); will give compile time error..

C: System.out.println(b.i); will print 20

D: None of the above statements are correct.

Q. No. 20

Question:

Given:

1. enum Animals {

2. DOG("woof"), CAT("meow"), FISH("burble");

3. String sound;

4. Animals(String s) { sound = s; }

5. }

6. class TestEnum {

7. static Animals a;

8. public static void main(String [] args) {

9. System.out.println(a.DOG.sound + " " + a.FISH.sound);

10. }

11. }

What is the result?

Answer Choices

A: Multiple compilation errors

B: woof burble

C: Compilation fails due to an error on line 3

D: Compilation fails due to an error on line 9

Q. No. 21

Question:

Given two files:

1. package pkgA;

2. public class Foo {

3. int a = 5;

4. protected int b = 6;

5. public int c = 7;

6. }

3. package pkgB;

4. import pkgA.\*;

5. public class Baz {

6. public static void main(String[] args) {

7. Foo f = new Foo();

8. System.out.print(" " + f.a);

9. System.out.print(" " + f.b);

10. System.out.print(" " + f.c);

11. }

12. }

What is the result?

Answer Choices

A: Compilation error at line 8 & 9

B: Compilation error at line 8

C: Compilation error at line 9

D: prints 5 6 7

Q. No. 22

Question:

Given:

class Clidder {

private final void flipper() { System.out.println("Clidder"); }

}

public class Clidlet extends Clidder {

public final void flipper() { System.out.println("Clidlet"); }

public static void main(String [] args) {

new Clidlet().flipper();

} }

What is the result?

Answer Choices

A: compilation error

B: Prints Clidder

C: Prints Clidlet

D: Prints Clidder

Cliddlet

Q. No. 23

Question:

What are all the methods used for Inter Thread communication and what is the class in which these methods are defined?

Answer Choices

A: wait(),notify() & notifyAll() belonging to Thread class.

B: wait(),notify() & notifyAll() belonging to Object class.

C: wait(),start() & run() belonging to Thread class.

D: interrupt(),join() & wakeup() belonging to Object class.

Q. No. 24

Question:

Given following 2 classes .

import java.util.\*;

class NameList {

List<String> names =new LinkedList<>();

public void add(String name) {

names.add(name);

}

public String removeFirst() {

if (names.size() > 0) {

try {

Thread.sleep(10);

} catch (InterruptedException e) {System.out.println(“error”);}

return names.remove(0);

} else

return null;

}

}

class TestSafeCollections {

public static void main(String[] args) {

final NameList nl = new NameList();

nl.add("abc");

class NameDropper extends Thread {

public void run() {

synchronized (nl) {

String name = nl.removeFirst();

System.out.println(name);

}

}

}

Thread t1 = new NameDropper();

Thread t2 = new NameDropper();

t1.start();

t2.start();

}

}

What will be the result?

Answer Choices

A: Prints abc

null

B: Prints abc & then raises IndexOutOfBoundsException

C: Prints error

D: Raises IllegalStateException

Q. No. 25

Question:

public class TestThreads {

public static void main(String[] args) {

Runnable r1=new Runnable() {

@Override

public void run() {

System.out.println("Invoked by +Thread.currentThread().getName());

}

};

Thread t1=new Thread(r1, "thread-1");

t1.run();

}

}

What will be the result ?

Answer Choices

A: Compiler error

B: Run time error

C: Prints “Invoked by thread-1”

D: None of above

Q. No. 26

Question:

Given :

class Utils {

synchronized void test()

{

System.out.println("entered test "+Thread.currentThread().getName());

try {

Thread.sleep(1000);

} catch (InterruptedException e) {}

System.out.println("exited test "+Thread.currentThread().getName());

}

synchronized void testMe()

{

System.out.println("entered testMe "+Thread.currentThread().getName());

try {

Thread.sleep(1000);

} catch (InterruptedException e) {}

System.out.println("exited test "+Thread.currentThread().getName());

}

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

final Utils u1 = new Utils();

Runnable r1 = new Runnable() {

@Override

public void run() { while (true)

u1.test(); }

};

Runnable r2 = new Runnable() {

@Override

public void run() {

while (true)

u1.testMe();

}

};

Thread t1 = new Thread(r1, "t1");

Thread t2 = new Thread(r2, "t2");

t1.start();

t2.start();

}

}

Answer Choices

A: Possible output

entered test t1

entered testMe t2

exited test t1

exited testMe t2

B: Possible output

entered testMe t2

entered test t1

exited testMe t2

exited test t1

C: Possible output

entered test t1

entered testMe t2

exited testMe t2

exited test t1

D: None of above

Q. No. 27

Question:

Given

import javax.swing.\*;

public class SwingFrame extends JFrame

{

public static void main(String[] args) {

JFrame f1=new SwingFrame();

}

public SwingFrame() {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

System.out.println(“Invoked by “+Thread.currentThread().getName());

}

});

}

}

Answer Choices

A: Run time error

B: No error , prints blank

C: Prints “Invoked by AWT-EventQueue-0”

D: Prints “Invoked by main”

Q. No. 28

Question:

What are the features of PrintWriter?

Answer Choices

A: Supplies buffering & automatic flushing of data.

B: Can wrap any binary as well as char stream

C: Supports print,println,printf API methods

D: All of above.

Q. No. 29

Question:

Which of the following statements is NOT true?

Answer Choices

A: If super class is serializable , sub-class is automatically serializable.

B: During de-serialization, default constructor is always called.

C: During de-serialization, binary stream containing state of the object & its corresponding Class object(Class<T>) are both essential

D: During serialization , transient & static data members are skipped from serialization.

Q. No. 30

Question:

Which of the following statements is true?

Answer Choices

A: Non static nested classes can not contain any static member.

B: Non static nested classes can not access outer class’s private members directly.

C: Static nested class can directly access non static member of outer class.

D: Method local inner class can access non final method local variable .

Q. No. 31

Question:

Given:

3. class SubException extends Exception { }

4. class SubSubException extends SubException { }

5.

6. public class CC { void doStuff() throws SubException { } }

7.

8. class CC2 extends CC { void doStuff() throws SubSubException { } }

9.

10. class CC3 extends CC { void doStuff() throws Exception { } }

11.

12. class CC4 extends CC { void doStuff(int x) throws Exception { } }

13.

14. class CC5 extends CC { void doStuff() { } }

What is the result?

Answer Choices

A: Compilation fails due to an error on line 8

B: Compilation fails due to an error on line 10

C: Compilation fails due to an error on line 12

D: No compiler error

Q. No. 32

Question:

Given:

3. public class Ebb {

4. static int x = 7;

5. public static void main(String[] args) {

6. String s = "";

7. for(int y = 0; y < 3; y++) {

8. x++;

9. switch(x) {

10. case 8: s += "8 ";

11. case 9: s += "9 ";

12. case 10: { s+= "10 "; break; }

13. default: s += "d ";

14. case 13: s+= "13 ";

15. }

16. }

17. System.out.println(s);

18. }

19. static { x++; }

20. }

What is the result?

Answer Choices

A: 8 9 10 d

B: 9 10 10 d

C: 9 10 10 d 13

D: 8 9 10 10 d 13

Q. No. 33

Question:

What is essential to make a deep clone copy of any object of user defined type eg Emp?

Answer Choices

A: The class must implement Cloneable interface

B: Must override clone() method properly

C: A & B both

D: Neither A nor B is essential

Q. No. 34

Question:

Given :

public class TestStrings

{

public static void main(String[] args)

{

String one = "someString";

String two = "someString";

String three=new String(one);

System.out.print(one.equals(two));

System.out.print(one == two);

System.out.print(one.equals(three));

System.out.print(one == three);

}

}

Answer Choices

A: true true true true

B: true false true false

C: true true false true

D: true true true false

Q. No. 35

Question:

Given

public class ImmutableStrings

{

public static void main(String[] args)

{

String one = "someString";

String two = new String("someString");

String three=”someString”;

one = two = three=null;

System.out.println(“testing ”);//1

System.out.println(“testing again ”);//2

}

}

At line no 1 , how many objects are marked for garbage collection?

Answer Choices

A: 0 B: 1

C: 2 D: 3

Q. No. 36

Question:

Consider following business requirement. Multiple bank branches need to upload suitable collection of bank accounts to a centralized server. Upon receiving collection server sends text acknowledgement to the clients & then saves this received collection to binary file . Assume that clients & server is developed using TCP sockets. Suggest suitable networking data streams that , a client application has to open for this reuirement.

Answer Choices

A: ObjectInputStream, BufferedReader

B: Scanner, PrintWriter

C: FileOutputStream, BufferedReader

D: ObjectOutputStream, BufferedReader

Q. No. 37

Question:

What is mandatory contract between overriding of hashCode & equals , for correct working of HashSet ?

Answer Choices

A: No such contract exists

B: If 2 objects are equal via equals method, then they must produce same integer hash codes.

C: If 2 objects are un equal via equals method, then they must produce distinct integer hash codes.

D: B & C both are mandatory

Q. No. 38

Question:

Which of following is NOT true?

Answer Choices

A: Final classes can not be extended.

B: Since abstract classes can’t be instantiated, they can’t contain any constructor.

C: Abstract class may contain complete concrete functionality.

D: Final references can not be re assigned.

Q. No. 39

Question:

Given :

import java.util.\*;

public class Test8 {

public static void main(String[] args) {

List<Integer> l1=Arrays.asList(12,1,234,145,100,45,87);

Collections.sort(l1, new Comparator<Integer>() {

@Override

public int compare(Integer o1, Integer o2) {

return o2-o1;

}});

System.out.println("List : "+l1);

}

}

What is the result ?

Answer Choices

A: Compiler error.

B: ClassCast Exception

C: List contents printed as per ascending order.

D: List contents printed as per descending order.

Q. No. 40

Question:

What will happen if a class extends Thread , does not override run() method & its object is created & invoked start() on the same object?

Answer Choices

A: Compiler Error

B: Run time error

C: Thread is permanently blocked

D: None of above