



# Oracle

## Exam Questions 1Z0-809

Java SE 8 Programmer II

### NEW QUESTION 1

Given:

```
class Sum extends RecursiveAction { //line n1 static final int THRESHOLD_SIZE = 3;
int stIndex, lstIndex; int [ ] data;
public Sum (int [ ]data, int start, int end) { this.data = data;
this stIndex = start; this. lstIndex = end;
}
protected void compute ( ) { int sum = 0;
if (lstIndex – stIndex <= THRESHOLD_SIZE) { for (int i = stIndex; i < lstIndex; i++) {
sum += data [i];
}
System.out.println(sum);
} else {
new Sum (data, stIndex + THRESHOLD_SIZE, lstIndex).fork( ); new Sum (data, stIndex,
Math.min (lstIndex, stIndex + THRESHOLD_SIZE)
).compute ();
}
}
}
```

and the code fragment:

```
ForkJoinPool fjPool = new ForkJoinPool ( ); int data [ ] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
fjPool.invoke (new Sum (data, 0, data.length));
and given that the sum of all integers from 1 to 10 is 55. Which statement is true?
```

- A. The program prints several values that total 55.
- B. The program prints 55.
- C. A compilation error occurs at line n1.
- D. The program prints several values whose sum exceeds 55.

**Answer:** A

### NEW QUESTION 2

Given the code fragments:

```
public class Test {
    List<String> list = null;
    public void printValues() {
        System.out.print (getList());
    }
    public List<String> getList(){ return list; }
    public void setList(List<String> newList){ list = newList; }
}
```

and

```
List<String> li = Arrays.asList("Dog", "Cat", "Mouse");
Test t = new Test();
t.setList(li.stream().collect(Collectors.toList()));
t.getList().forEach(Test::printValues);
```

What is the result?

- A. null
- B. A compilation error occurs.
- C. DogCatMouse
- D. [Dog, Cat, Mouse]

**Answer:** D

### NEW QUESTION 3

Given the code fragment:

```
public class FileThread implements Runnable { String fName;
public FileThread(String fName) { this.fName = fName; } public void run () System.out.println(fName);}
public static void main (String[] args) throws IOException, InterruptedException {
ExecutorService executor = Executors.newCachedThreadPool(); Stream<Path> listOfFiles = Files.walk(Paths.get("Java Projects")); listOfFiles.forEach(line -> {
executor.execute(new FileThread(line.getFileName().toString ()); //
line n1
});
executor.shutdown(); executor.awaitTermination(5, TimeUnit.DAYS); // line n2
}
}
```

The Java Projects directory exists and contains a list of files. What is the result?

- A. The program throws a runtime exception at line n2.

- B. The program prints files names concurrently.
- C. The program prints files names sequentially.
- D. A compilation error occurs at line n1.

**Answer:** B

#### NEW QUESTION 4

Given the code fragment:

```
List<String> codes = Arrays.asList ("DOC", "MPEG", "JPEG"); codes.forEach (c -> System.out.print(c + " "));  
String fmt = codes.stream()  
.filter (s-> s.contains ("PEG"))  
.reduce((s, t) -> s + t).get(); System.out.println("\n" + fmt); What is the result?
```

- A. DOC MPEG JPEG MPEGJPEG
- B. DOC MPEG MPEGJPEG MPEGMPEGJPEG
- C. MPEGJPEG MPEGJPEG
- D. The order of the output is unpredictable.

**Answer:** A

#### NEW QUESTION 5

Which class definition compiles?

```
A. class Vehicle {  
    int id;  
    public void start() {  
        public class Engine {    int eNo = id;    }  
    }  
}  
  
B. class Computer {  
    private Card sCard = new SoundCard();  
    private abstract class Card { }  
    private class SoundCard extends Card { }  
}  
  
C. class Block {  
    int bno;  
    static class Counter {  
        int locator;  
        Counter() { locator = bno; }  
    }  
}  
  
D. class Product {  
    interface Moveable { void move(); }  
    Moveable mProduct = new Moveable() {  
        void move() { }  
    };  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

#### NEW QUESTION 6

Given:

```
public class Foo {
    public void methodB(String s) { System.out.println("Foo " + s ); }
}

public class Bar extends Foo {
    public void methodB(String s) { System.out.println("Bar " + s); }
}

public class Baz extends Bar {
    public void methodB(String s) { System.out.println("Baz " + s); }
}

public class Daze extends Baz{
    private Bar bb = new Bar();
    public void methodB(String s) {
        bb.methodB(s);
        super.methodB(s);
    }
}

public class TestClass {
    public static void main(String[] args) {
        Baz d = new Daze();
        d.methodB("Hello");
    }
}
```

What is the result?

- A. Bar Hello Foo Hello
- B. Bar Hello Baz Hello
- C. Baz Hello
- D. A compilation error occurs in the Daze class.

**Answer: C**

#### NEW QUESTION 7

Given the code fragment:

```
List<String> cs = Arrays.asList("Java", "Java EE", "Java ME");
// line n1
System.out.print(b);
```

Which code fragment, when inserted at line n1, ensures false is printed?

- A. boolean b = cs.stream().findAny().get().equals("Java");
- B. boolean b = cs.stream().anyMatch(w -> w.equals("Java"));
- C. boolean b = cs.stream().findFirst().get().equals("Java");
- D. boolean b = cs.stream().allMatch(w -> w.equals("Java"));

**Answer: C**

#### NEW QUESTION 8

Given the code fragment:

```
List<String> listVal = Arrays.asList("Joe", "Paul", "Alice", "Tom"); System.out.println (
// line n1
);
```

Which code fragment, when inserted at line n1, enables the code to print the count of string elements whose length is greater than three?

- A. listVal.stream().filter(x -> x.length()>3).count()
- B. listVal.stream().map(x -> x.length()>3).count()
- C. listVal.stream().peek(x -> x.length()>3).count().get()
- D. listVal.stream().filter(x -> x.length()>3).mapToInt(x -> x).count()

**Answer: A**

#### NEW QUESTION 9

Given the content:

```
MessagesBundle.properties file:

inquiry = How are you?

MessagesBundle_de_DE.properties file:

inquiry = Wie geht's?
```

and given the code fragment:

```
Locale currentLocale;
// line 1
ResourceBundle messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);
System.out.println(messages.getString("inquiry"));
```

Which two code fragments, when inserted at line 1 independently, enable the code to print “Wie geht’s?”

- A. currentLocale = new Locale (“de”, “DE”);
- B. currentLocale = new Locale.Builder ().setLanguage (“de”).setRegion (“DE”).build ();
- C. currentLocale = Locale.GERMAN;
- D. currentLocale = new Locale(); currentLocale.setLanguage (“de”); currentLocale.setRegion (“DE”);
- E. currentLocale = Locale.getInstance(Locale.GERMAN,Locale.GERMANY);

**Answer: B**

#### NEW QUESTION 10

Given:

```
public class Foo<K, V> {
    private K key;
    private V value;

    public Foo(K key, V value) { this.key = key; this.value = value; }

    public static <T> Foo<T, T> twice(T value) { return new Foo<T, T>(value, value); }

    public K getKey() { return key; }
    public V getValue() { return value; }
}
```

Which option fails?

- A. Foo<String, Integer> mark = new Foo<String, Integer> (“Steve”, 100);
- B. Foo<String, String> pair = Foo.<String>twice (“Hello World!”);
- C. Foo<Object, Object> percentage = new Foo<String, Integer>(“Steve”, 100);
- D. Foo<String, String> grade = new Foo <> (“John”, “A”);

**Answer: A**

#### NEW QUESTION 10

Given:

```
IntStream stream = IntStream.of (1,2,3); IntFunction<Integer> inFu= x -> y -> x*y; //line n1
IntStream newStream = stream.map(inFu.apply(10)); //line n2 newStream.forEach(System.out::print);
```

Which modification enables the code fragment to compile?

- A. Replace line n1 with: IntFunction<UnaryOperator> inFu = x -> y -> x\*y;
- B. Replace line n1 with: IntFunction<IntUnaryOperator> inFu = x -> y -> x\*y;
- C. Replace line n1 with: BiFunction<IntUnaryOperator> inFu = x -> y -> x\*y;
- D. Replace line n2 with: IntStream newStream = stream.map(inFu.applyAsInt (10));

**Answer: B**

#### NEW QUESTION 12

Given:

```
public class Customer { private String fName; private String lName; private static int count;
public customer (String first, String last) {fName = first, lName = last;
++count;}
static { count = 0; }
public static int getCount() {return count; }
}

public class App {
public static void main (String [] args) { Customer c1 = new Customer(“Larry”, “Smith”);
Customer c2 = new Customer(“Pedro”, “Gonzales”); Customer c3 = new Customer(“Penny”, “Jones”); Customer c4 = new Customer(“Lars”, “Svenson”); c4 =
```

```
null;
c3 = c2;
System.out.println (Customer.getCount());
}
}
```

What is the result?

- A. 2
- B. 3
- C. 4
- D. 5

**Answer: D**

#### NEW QUESTION 17

Given:

```
class Bird {
public void fly () { System.out.print("Can fly"); }
}
class Penguin extends Bird {
public void fly () { System.out.print("Cannot fly"); }
}
```

and the code fragment: class Birdie {  
public static void main (String [ ] args) { fly( ( ) -> new Bird ( ));  
fly (Penguin : : new);  
}  
/\* line n1 \*/  
}

Which code fragment, when inserted at line n1, enables the Birdie class to compile?

- A. static void fly (Consumer<Bird> bird) { bird :: fly ();}
- B. static void fly (Consumer<? extends Bird> bird) {bird.accept( ) fly ();}
- C. static void fly (Supplier<Bird> bird) { bird.get( ) fly ();}
- D. static void fly (Supplier<? extends Bird> bird) { LOST

**Answer: C**

#### NEW QUESTION 19

Given the code fragment:

```
public void recDelete (String dirName) throws IOException { File [ ] listOfFiles = new File (dirName) .listFiles();
if (listOfFiles != null && listOfFiles.length >0) {
for (File aFile : listOfFiles) { if (aFile.isDirectory ()) {
recDelete (aFile.getAbsolutePath ());
} else {
if (aFile.getName ().endsWith (".class")) aFile.delete ();
}
}
}
}
```

Assume that Projects contains subdirectories that contain .class files and is passed as an argument to the recDelete () method when it is invoked. What is the result?

- A. The method deletes all the .class files in the Projects directory and its subdirectories.
- B. The method deletes the .class files of the Projects directory only.
- C. The method executes and does not make any changes to the Projects directory.
- D. The method throws an IOException.

**Answer: A**

#### NEW QUESTION 20

What is true about the java.sql.Statement interface?

- A. It provides a session with the database.
- B. It is used to get an instance of a Connection object by using JDBC drivers.
- C. It provides a cursor to fetch the resulting data.
- D. It provides a class for executing SQL statements and returning the results.

**Answer: D**

#### NEW QUESTION 23

Given:

```
1. abstract class Shape {
2. Shape ( ) { System.out.println ("Shape"); }
3. protected void area ( ) { System.out.println ("Shape"); } 4. }
5.
6. class Square extends Shape {
7. int side;
8. Square int side {
9. /* insert code here */
```

```
10. this.side = side;
11. }
12. public void area ( ) { System.out.println ("Square"); }
13. }
14. class Rectangle extends Square {
15. int len, br;
16. Rectangle (int x, int y) {
17. /* insert code here */
18. len = x, br = y;
19. }
20. void area ( ) { System.out.println ("Rectangle"); }
21. }
```

Which two modifications enable the code to compile? (Choose two.)

- A. At line 1, remove abstract
- B. At line 9, insert super ( );
- C. At line 12, remove public
- D. At line 17, insert super (x);
- E. At line 17, insert super (); super.side = x;
- F. At line 20, use public void area ( ) {

**Answer:** DF

#### NEW QUESTION 26

Given the definition of the Emp class: public class Emp  
private String eName; private Integer eAge;  
Emp(String eN, Integer eA) { this.eName = eN;  
this.eAge = eA;  
}  
public Integer getEAge () {return eAge;} public String getENAME () {return eName;}  
}

and code fragment:

```
List<Emp>li = Arrays.asList(new Emp("Sam", 20), New Emp("John", 60), New Emp ("Jim", 51));
Predicate<Emp> agVal = s -> s.getEAge() > 50; //line n1 li = li.stream().filter(agVal).collect(Collectors.toList());
Stream<String> names = li.stream().map.(Emp::getENAME); //line n2 names.forEach(n -> System.out.print(n + " ");
What is the result?
```

- A. Sam John Jim
- B. John Jim
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

**Answer:** B

#### NEW QUESTION 27

Given the records from the Employee table:

eid	ename
111	Tom
112	Jerry
113	Donald

and given the code fragment: try {  
Connection conn = DriverManager.getConnection (URL, userName, passWord); Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,  
ResultSet.CONCUR\_UPDATABLE);  
st.execute("SELECT\*FROM Employee"); ResultSet rs = st.getResultSet();  
while (rs.next()) {  
if (rs.getInt(1) ==112) { rs.updateString(2, "Jack");  
}  
}  
rs.absolute(2);  
System.out.println(rs.getInt(1) + " " + rs.getString(2));  
} catch (SQLException ex) { System.out.println("Exception is raised");  
}  
}

Assume that:

The required database driver is configured in the classpath.

The appropriate database accessible with the URL, userName, and passWord exists. What is the result?

- A. The Employee table is updated with the row: 112 Jackand the program prints: 112 Jerry
- B. The Employee table is updated with the row: 112 Jackand the program prints: 112 Jack
- C. The Employee table is not updated and the program prints: 112 Jerry
- D. The program prints Exception is raised.

**Answer:** A

#### NEW QUESTION 29

You want to create a singleton class by using the Singleton design pattern. Which two statements enforce the singleton nature of the design? (Choose two.)

- A. Make the class static.

- B. Make the constructor private.
- C. Override equals() and hashCode() methods of the java.lang.Object class.
- D. Use a static reference to point to the single instance.
- E. Implement the Serializable interface.

**Answer:** BD

### NEW QUESTION 33

Given the code fragment:

```
BiFunction<Integer, Double, Integer> val = (t1, t2) -> t1 + t2; //line n1 System.out.println(val.apply(10, 10.5));
```

What is the result?

- A. 20
- B. 20.5
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

**Answer:** C

### NEW QUESTION 37

Given:

```
class FuelNotAvailException extends Exception { } class Vehicle {  
void ride() throws FuelNotAvailException { //line n1 System.out.println("Happy Journey!");  
}  
}
```

```
class SolarVehicle extends Vehicle {  
public void ride () throws Exception { //line n2 super ride ();  
}  
}
```

and the code fragment:

```
public static void main (String[] args) throws FuelNotAvailException, Exception  
{  
Vehicle v = new SolarVehicle (); v.ride();  
}
```

Which modification enables the code fragment to print Happy Journey!?

- A. Replace line n1 with public void ride() throws FuelNotAvailException {
- B. Replace line n1 with protected void ride() throws Exception {
- C. Replace line n2 with void ride() throws Exception {
- D. Replace line n2 with private void ride() throws FuelNotAvailException {

**Answer:** B

### NEW QUESTION 39

Given the code fragments:

```
public class Book implements Comparator<Book> { String name;  
double price; public Book () {}  
public Book(String name, double price) { this.name = name;  
this.price = price;  
}  
public int compare(Book b1, Book b2) { return b1.name.compareTo(b2.name);  
}  
public String toString() { return name + ":" + price;  
}  
}
```

and

```
List<Book>books = Arrays.asList (new Book ("Beginning with Java", 2), new book ("A  
Guide to Java Tour", 3));  
Collections.sort(books, new Book()); System.out.print(books);
```

What is the result?

- A. [A Guide to Java Tour:3.0, Beginning with Java:2.0]
- B. [Beginning with Java:2, A Guide to Java Tour:3]
- C. A compilation error occurs because the Book class does not override the abstract method compareTo().
- D. An Exception is thrown at run time.

**Answer:** A

### NEW QUESTION 43

Given that course.txt is accessible and contains:

Course : : Java

and given the code fragment:

```
public static void main (String[ ] args) { int i;  
char c;  
try (FileInputStream fis = new FileInputStream ("course.txt"); InputStreamReader isr = new InputStreamReader(fis);) { while (isr.ready()) { //line n1  
isr.skip(2);  
i = isr.read (); c = (char) i;  
System.out.print(c);  
}  
}
```

```
} catch (Exception e) { e.printStackTrace();  
}  
}
```

What is the result?

- A. ur :: va
- B. ueJa
- C. The program prints nothing.
- D. A compilation error occurs at line n1.

**Answer: B**

#### NEW QUESTION 44

Given:

```
interface P { public void method1(); }  
  
interface Q extends P { public void method1(); }  
  
interface R extends P { public void method2(); }  
  
interface S { public default void method() { } }  
  
interface T { public void method1(); public void method2(); }  
  
interface U { public void method1(); public abstract void method2(); }
```

Which two interfaces can you use to create lambda expressions? (Choose two.)

- A. T
- B. R
- C. P
- D. S
- E. Q
- F. U

**Answer: AF**

#### NEW QUESTION 46

In 2015, daylight saving time in New York, USA, begins on March 8th at 2:00 AM. As a result, 2:00 AM becomes 3:00 AM.

Given the code fragment:

```
ZoneId zone = ZoneId.of("America/New_York");  
ZonedDateTime dt = ZonedDateTime.of(LocalDate.of(2015, 3, 8), LocalTime.of(1, 0),  
zone);  
ZonedDateTime dt2 = dt.plusHours(2);  
System.out.print(DateTimeFormatter.ofPattern("H:mm - ").format(dt2));  
System.out.println("difference: " + ChronoUnit.HOURS.between(dt, dt2));
```

Which is the result?

- A. 3:00 – difference: 2
- B. 2:00 – difference: 1
- C. 4:00 – difference: 3
- D. 4:00 – difference: 2

**Answer: B**

#### NEW QUESTION 48

Given:

```
interface Interfacel {
    public default void sayHi() {
        System.out.println("Hi Interface-1");
    }
}

interface Interface2 {
    public default void sayHi() {
        System.out.println("Hi Interface-2");
    }
}

public class MyClass implements Interfacel, Interface2 {
    public static void main(String[] args) {
        Interfacel obj = new MyClass();
        obj.sayHi();
    }
    public void sayHi() {
        System.out.println("Hi MyClass");
    }
}
```

What is the result?

- A. Hi Interface-2
- B. A compilation error occurs.
- C. Hi Interface-1
- D. Hi MyClass

**Answer:** D

#### NEW QUESTION 52

Given the code fragment:

Path source = Paths.get ("/data/december/log.txt"); Path destination = Paths.get("/data");

Files.copy (source, destination);

and assuming that the file /data/december/log.txt is accessible and contains: 10-Dec-2014 – Executed successfully

What is the result?

- A. A file with the name log.txt is created in the /data directory and the content of the /data/december/ log.txt file is copied to it.
- B. The program executes successfully and does NOT change the file system.
- C. A FileNotFoundException is thrown at run time.
- D. A FileAlreadyExistsException is thrown at run time.

**Answer:** D

#### NEW QUESTION 53

Given:

```
public interface LengthValidator {
    public boolean checkLength(String str);
}
```

and

```
public class Txt {
    public static void main(String[] args) {
        boolean res = new LengthValidator() {
            public boolean checkLength(String str) {
                return str.length() > 5 && str.length() < 10;
            }
        }.checkLength("Hello");
    }
}
```

Which interface from the java.util.function package should you use to refactor the class Txt?

- A. Consumer

- B. Predicate
- C. Supplier
- D. Function

**Answer:** C

#### NEW QUESTION 55

Given the code fragment: `Stream<List<String>> iStr= Stream.of ( Arrays.asList ("1", "John"), Arrays.asList ("2", null)0; Stream<<String> nInSt = iStr.flatMapToInt ((x) -> x.stream ()); nInSt.forEach (System.out :: print);`  
What is the result?

- A. 1John2null
- B. 12
- C. A `NullPointerException` is thrown at run time.
- D. A compilation error occurs.

**Answer:** D

#### NEW QUESTION 58

Given the code fragment:  
`List<String> colors = Arrays.asList("red", "green", "yellow"); Predicate<String> test = n -> { System.out.println("Searching..."); return n.contains("red"); };`  
`colors.stream().filter(c -> c.length() > 3).allMatch(test);` What is the result?

- A. Searching...
- B. Searching...Searching...
- C. Searching...Searching... Searching...
- D. A compilation error occurs.

**Answer:** A

#### NEW QUESTION 63

Given:  
`interface Rideable {Car getCar (String name); } class Car { private String name; public Car (String name) { this.name = name; } }`

Which code fragment creates an instance of `Car`?

- A. `Car auto = Car ("MyCar"): : new;`
- B. `Car auto = Car : : new;Car vehicle = auto : : getCar("MyCar");`
- C. `Rideable rider = Car : : new;Car vehicle = rider.getCar("MyCar");`
- D. `Car vehicle = Rideable : : new : : getCar("MyCar");`

**Answer:** C

#### NEW QUESTION 65

Given the code fragment:  
`List<Integer> nums = Arrays.asList (10, 20, 8); System.out.println ( //line n1 );`  
Which code fragment must be inserted at line n1 to enable the code to print the maximum number in the `nums` list?

- A. `nums.stream().max(Comparator.comparing(a -> a)).get()`
- B. `nums.stream().max(Integer : : max).get()`
- C. `nums.stream().max()`
- D. `nums.stream().map(a -> a).max()`

**Answer:** A

#### NEW QUESTION 68

Given the code fragment:

```
List<Integer> prices = Arrays.asList(3, 4, 5);
prices.stream()
    .filter(e -> e > 4)
    .peek(e -> System.out.print("Price " + e))           // line n1
    .map(n -> n - 1)                                     // line n2
    .peek(n -> System.out.println(" New Price " + n));    // line n3
```

Which modification enables the code to print `Price 5 New Price 4`?

- A. Replace line n2 with `.map (n -> System.out.println ("New Price" + n –1))` and remove line n3

- B. Replace line n2 with `.mapToInt (n -> n - 1);`
- C. Replace line n1 with `.forEach (e -> System.out.print ("Price" + e))`
- D. Replace line n3 with `.forEach (n -> System.out.println ("New Price" + n));`

**Answer:** A

#### NEW QUESTION 71

Given:

```
class RateOfInterest {  
    public static void main (String[] args) { int rateOfInterest = 0;  
    String accountType = "LOAN"; switch (accountType) {  
    case "RD"; rateOfInterest = 5; break;  
    case "FD"; rateOfInterest = 10; break;  
    default:  
    assert false: "No interest for this account"; //line n1  
    }  
    System.out.println ("Rate of interest:" + rateOfInterest);  
    }  
}
```

and the command:

`java -ea RateOfInterest` What is the result?

- A. Rate of interest: 0
- B. An AssertionError is thrown.
- C. No interest for this account
- D. A compilation error occurs at line n1.

**Answer:** B

#### NEW QUESTION 76

Given the code fragment:

```
Deque<String> queue = new ArrayDeque<>();  
queue.add("Susan");  
queue.add("Allen");  
queue.add("David");  
System.out.println(queue.pop());  
System.out.println(queue.remove());  
System.out.println(queue);
```

What is the result?

- A. DavidDavid[Susan, Allen]
- B. SusanSusan[Susan, Allen]
- C. SusanAllen [David]
- D. DavidAllen [Susan]
- E. SusanAllen[Susan, David]

**Answer:** C

#### NEW QUESTION 81

Given the code fragment:

```
// Login time:2015-01-12T21:58:18.817Z  
Instant loginTime = Instant.now();  
Thread.sleep(1000);  
  
// Logout time:2015-01-12T21:58:19.880Z  
Instant logoutTime = Instant.now();  
  
loginTime = loginTime.truncatedTo(ChronoUnit.MINUTES); // line n1  
logoutTime = logoutTime.truncatedTo(ChronoUnit.MINUTES);  
  
if (logoutTime.isAfter(loginTime))  
    System.out.println("Logged out at:"+logoutTime);  
else  
    System.out.println("Can't logout");
```

What is the result?

- A. A compilation error occurs at line n1.

- B. Logged out at: 2015-01-12T21:58:19.880Z
- C. Can't logout
- D. Logged out at: 2015-01-12T21:58:00Z

**Answer:** D

#### NEW QUESTION 85

Assume customers.txt is accessible and contains multiple lines. Which code fragment prints the contents of the customers.txt file?

- A. `Stream<String> stream = Files.find (Paths.get ("customers.txt")); stream.forEach((String c) -> System.out.println(c));`
- B. `Stream<Path> stream = Files.find (Paths.get ("customers.txt")); stream.forEach( c) -> System.out.println(c));`
- C. `Stream<Path> stream = Files.list (Paths.get ("customers.txt")); stream.forEach( c) -> System.out.println(c));`
- D. `Stream<String> lines = Files.lines (Paths.get ("customers.txt")); lines.forEach( c) -> System.out.println(c));`

**Answer:** A

#### NEW QUESTION 86

Given the code fragment:

```
Map<Integer, Integer> mVal = new HashMap<>();  
mVal.put(1, 10);  
mVal.put(2, 20);  
//line n1  
c.accept(1, 2);  
mVal.forEach(c);
```

Which statement can be inserted into line n1 to print 1,2; 1,10; 2,20;?

- A. `BiConsumer<Integer,Integer> c = (i, j) -> {System.out.print (i + "," + j+ "; ");}`
- B. `BiFunction<Integer, Integer, String> c = (i, j) -> {System.out.print (i + "," + j+ "; ");}`
- C. `BiConsumer<Integer, Integer, String> c = (i, j) -> {System.out.print (i + "," + j+ "; ");}`
- D. `BiConsumer<Integer, Integer, Integer> c = (i, j) -> {System.out.print (i + ","+ j+ "; ");}`

**Answer:** B

#### NEW QUESTION 88

Given:

```
class Vehicle { int vno;  
String name;  
public Vehicle (int vno, String name) { this.vno = vno,;  
this.name = name;  
}  
public String toString () { return vno + ":" + name;  
}  
}
```

and this code fragment:

```
Set<Vehicle> vehicles = new TreeSet <> (); vehicles.add(new Vehicle (10123, "Ford")); vehicles.add(new Vehicle (10124, "BMW")); System.out.println(vehicles);
```

What is the result?

- A. 10123 Ford10124 BMW
- B. 10124 BMW10123 Ford
- C. A compilation error occurs.
- D. A ClassCastException is thrown at run time.

**Answer:** D

#### NEW QUESTION 91

Given the code fragment:

```
Map<Integer, String> books = new TreeMap<>(); books.put (1007, "A");  
books.put (1002, "C");  
books.put (1001, "B");  
books.put (1003, "B"); System.out.println (books); What is the result?
```

- A. {1007 = A, 1002 = C, 1001 = B, 1003 = B}
- B. {1001 = B, 1002 = C, 1003 = B, 1007 = A}
- C. {1002 = C, 1003 = B, 1007 = A}
- D. {1007 = A, 1001 = B, 1003 = B, 1002 = C}

**Answer:** B

#### NEW QUESTION 96

Given the records from the STUDENT table:

sid	sname	semail
111	James	james@uni.com
112	Jane	jane@uni.com
114	John	john@uni.com

Given the code fragment:

```
public static void main(String[] args) throws SQLException {
    //code to load and register valid jdbc driver go here
    Connection con = DriverManager.getConnection(URL, username, password);
    Statement st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
                                       ResultSet.CONCUR_UPDATABLE);

    st.execute("SELECT * FROM student");
    ResultSet rs = st.getResultSet();
    rs.absolute(3);
    rs.moveToInsertRow();
    rs.updateInt(1, 113);
    rs.updateString(2, "Jannet");
    rs.updateString(3, "jannet@uni.com");
    rs.updateRow();
    rs.refreshRow();
    System.out.println(rs.getInt(1) + " : " + rs.getString(2) + " : " + rs.getString
(3));
}
```

Assume that the URL, username, and password are valid. What is the result?

- A. The STUDENT table is not updated and the program prints: 114 : John : john@uni.com
- B. The STUDENT table is updated with the record: 113 : Jannet : jannet@uni.com and the program prints: 114 : John : john@uni.com
- C. The STUDENT table is updated with the record: 113 : Jannet : jannet@uni.com and the program prints: 113 : Jannet : jannet@uni.com
- D. A SQLException is thrown at run time.

**Answer:** A

#### NEW QUESTION 101

Given the code fragments:

```
public class Video {
    public void play() throws IOException {
        System.out.print("Video played.");
    }
}

public class Game extends Video {
    public void play() throws Exception {
        super.play();
        System.out.print("Game played.");
    }
}
```

and

```
try {
    new Game().play();
} catch (Exception e) {
    System.out.print(e.getClass());
}
```

What is the result?

- A. Video played.Game played.
- B. A compilation error occurs.
- C. class java.lang.Exception
- D. class java.io.IOException

Answer: C

#### NEW QUESTION 106

Given:

```
class Engine {
    double fuelLevel;
    Engine(int fuelLevel) { this.fuelLevel = fuelLevel; }
    public void start() {
        // line n1
        System.out.println("Started");
    }
    public void stop() { System.out.println("Stopped"); }
}
```

Your design requires that:

- ☒ fuelLevel of Engine must be greater than zero when the start() method is invoked.
- ☒ The code must terminate if fuelLevel of Engine is less than or equal to zero.

Which code fragment should be added at line n1 to express this invariant condition?

- A. `assert (fuelLevel) : "Terminating..."`;
- B. `assert (fuelLevel > 0) : System.out.println ("Impossible fuel")`;
- C. `assert fuelLevel < 0: System.exit(0)`;
- D. `assert fuelLevel > 0: "Impossible fuel"` ;

Answer: C

#### NEW QUESTION 110

Given the code fragment:

```
List<Integer> list1 = Arrays.asList(10, 20); List<Integer> list2 = Arrays.asList(15, 30);
//line n1
```

Which code fragment, when inserted at line n1, prints 10 20 15 30?

- A. `Stream.of(list1, list2).flatMap(list -> list.stream()).forEach(s -> System.out.print(s + " "));`
- B. `Stream.of(list1, list2).flatMap(list -> list.intStream()).forEach(s -> System.out.print(s + " "));`
- C. `list1.stream().flatMap(list2.stream()).flatMap(e1 -> e1.stream()).forEach(s -> System.out.println(s + " "));`
- D. `Stream.of(list1, list2).flatMapToInt(list -> list.stream()).forEach(s -> System.out.print(s + " "));`

Answer: A

#### NEW QUESTION 115

For which three objects must a vendor provide implementations in its JDBC driver? (Choose three.)

- A. Time
- B. Date
- C. Statement
- D. ResultSet
- E. Connection
- F. SQLException
- G. DriverManager

Answer: CDE

#### Explanation:

Database vendors support JDBC through the JDBC driver interface or through the ODBC connection. Each driver must provide implementations of `java.sql.Connection`, `java.sql.Statement`, `java.sql.PreparedStatement`, `java.sql.CallableStatement`, and `java.sql.ResultSet`. They must also implement the `java.sql.Driver` interface for use by the generic `java.sql.DriverManager` interface.

#### NEW QUESTION 120

Given the code fragment:

```
List<String> qwords = Arrays.asList("why ", "what ", "when ");
BinaryOperator<String> operator = (s1, s2) -> s1.concat(s2); // line n1
String sen = qwords.stream()
    .reduce("Word: ", operator);
System.out.println(sen);
```

What is the result?

- A. Word: why what when
- B. Word: why Word: why what Word: why what when
- C. Word: why Word: what Word: when
- D. Compilation fails at line n1.

**Answer:** A

#### NEW QUESTION 125

Given the code fragment:

```
9. Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
10. String query = "SELECT id FROM Employee";
11. try (Statement stmt = conn.createStatement()) {
12.     ResultSet rs = stmt.executeQuery(query);
13.     stmt.executeQuery("SELECT id FROM Customer");
14.     while (rs.next()) {
15.         //process the results
16.         System.out.println("Employee ID: "+ rs.getInt("id"));
17.     }
18. } catch (Exception e) {
19.     System.out.println ("Error");
20. }
```

Assume that:

The required database driver is configured in the classpath.

The appropriate database is accessible with the dbURL, userName, and passWord exists.

The Employee and Customer tables are available and each table has id column with a few records and the SQL queries are valid.

What is the result of compiling and executing this code fragment?

- A. The program prints employee IDs.
- B. The program prints customer IDs.
- C. The program prints Error.
- D. compilation fails on line 13.

**Answer:** C

#### NEW QUESTION 130

Given the code fragment:

```
List<String> empDetails = Arrays.asList("100, Robin, HR", "200, Mary, AdminServices",
"101, Peter, HR");
empDetails.stream()
    .filter(s-> s.contains("1"))
    .sorted()
    .forEach(System.out::println); //line n1
```

What is the result?

- A. 100, Robin, HR101, Peter, HR
- B. A compilation error occurs at line n1.
- C. 100, Robin, HR101, Peter, HR200, Mary, AdminServices
- D. 100, Robin, HR200, Mary, AdminServices101, Peter, HR

**Answer:** A

#### NEW QUESTION 131

Given the code fragment:

```
List<String> li = Arrays.asList("Java", "J2EE", "J2ME", "JSTL", "JSP", "Oracle DB");
Predicate<String> val = p -> p.contains("J");
List<String> neLi = li.stream().filter(x -> x.length() > 3)
    .filter(val).collect(Collectors.toList());
System.out.println(neLi);
```

What is the result?

- A. A compilation error occurs.
- B. [Java, J2EE, J2ME, JSTL, JSP]
- C. null
- D. [Java, J2EE, J2ME, JSTL]

**Answer:** A

#### NEW QUESTION 136

Given the code fragment:

```
List<String> nums = Arrays.asList("EE", "SE");
String ans = nums
    .parallelStream()
    .reduce("Java ", (a, b) -> a.concat(b));
System.out.print(ans);
```

What is the result?

- A. Java EEJava EESE
- B. Java EESE

- C. The program prints either:Java EEJava SE orJava SEJava EE  
D. Java EEJava SE

**Answer:** D

#### NEW QUESTION 140

Given the code fragments:

```
class Caller implements Callable<String> { String str;  
public Caller (String s) {this.str=s;}  
public String call()throws Exception { return str.concat ("Caller");}  
}  
class Runner implements Runnable { String str;  
public Runner (String s) {this.str=s;}  
public void run () { System.out.println (str.concat ("Runner"));}  
}  
and  
public static void main (String[] args) InterruptedException, ExecutionException  
{  
ExecutorService es = Executors.newFixedThreadPool(2); Future f1 = es.submit (new Caller ("Call"));  
Future f2 = es.submit (new Runner ("Run")); String str1 = (String) f1.get();  
String str2 = (String) f2.get(); //line n1 System.out.println(str1+ ":" + str2);  
}  
}
```

What is the result?

- A. The program prints: Run RunnerCall Caller : nullAnd the program does not terminate.  
B. The program terminates after printing: Run RunnerCall Caller : Run  
C. A compilation error occurs at line n1.  
D. An Execution is thrown at run time.

**Answer:** A

#### NEW QUESTION 145

Given:

```
class UserException extends Exception { }  
class AgeOutOfLimitException extends UserException { } and the code fragment:  
class App {  
public void doRegister(String name, int age) throws UserException, AgeOutOfLimitException { if (name.length () < 6) {  
throw new UserException ();  
} else if (age >= 60) {  
throw new AgeOutOfLimitException ();  
} else {  
System.out.println("User is registered.");  
}  
}  
}  
public static void main(String[] args) throws UserException { App t = new App ();
```

- A. t.doRegister("Mathew", 60);}}What is the result?  
B. User is registered.  
C. An AgeOutOfLimitException is thrown.  
D. A UserException is thrown.  
E. A compilation error occurs in the main method.

**Answer:** B

#### NEW QUESTION 148

Which statement is true about the single abstract method of the java.util.function.Function interface?

- A. It accepts one argument and returns void.  
B. It accepts one argument and returns boolean.  
C. It accepts one argument and always produces a result of the same type as the argument.  
D. It accepts an argument and produces a result of any data type.

**Answer:** D

#### NEW QUESTION 153

Given the code fragment:

```
List<String> str = Arrays.asList ("my", "pen", "is", "your", "pen"); Predicate<String> test = s -> {  
int i = 0;  
boolean result = s.contains ("pen");  
System.out.print(i++) + ":"; return result;  
};  
str.stream()  
.filter(test)  
.findFirst()  
.ifPresent(System.out ::print); What is the result?
```

- A. 0 : 0 : pen  
B. 0 : 1 : pen  
C. 0 : 0 : 0 : 0 : 0 : pen

- D. 0 : 1 : 2 : 3 : 4 :  
E. A compilation error occurs.

**Answer:** A

#### NEW QUESTION 157

Given:

```
public class Emp { String fName; String lName;  
public Emp (String fn, String ln) { fName = fn;  
lName = ln;  
}  
public String getfName() { return fName; } public String getlName() { return lName; }  
}
```

and the code fragment: List<Emp> emp = Arrays.asList ( new Emp ("John", "Smith"),  
new Emp ("Peter", "Sam"),  
new Emp ("Thomas", "Wale")); emp.stream()  
//line n1

.collect(Collectors.toList());

Which code fragment, when inserted at line n1, sorts the employees list in descending order of fName and then ascending order of lName?

- A. .sorted (Comparator.comparing(Emp::getfName).reversed().thenComparing(Emp::getlName))  
B. .sorted (Comparator.comparing(Emp::getfName).thenComparing(Emp::getlName))  
C. .map(Emp::getfName).sorted(Comparator.reverseOrder())  
D. .map(Emp::getfName).sorted(Comparator.reverseOrder()).map (Emp::getlName).reversed

**Answer:** A

#### NEW QUESTION 159

Given:

```
public class Test<T> { private T t;  
public T get () { return t;  
}  
public void set (T t) { this.t = t;  
}  
public static void main (String args [ ] ) { Test<String> type = new Test<>();  
Test type 1 = new Test (); //line n1 type.set("Java");  
type1.set(100); //line n2 System.out.print(type.get() + " " + type1.get());  
}  
}
```

What is the result?

- A. Java 100  
B. java.lang.string@<hashcode>java.lang.Integer@<hashcode>  
C. A compilation error occur  
D. To rectify it, replace line n1 with: Test<Integer> type1 = new Test<>();  
E. A compilation error occur  
F. To rectify it, replace line n2 with: type1.set (Integer(100));

**Answer:** A

#### NEW QUESTION 163

Given:

```
class Product {  
    String pname;  
    public Product(String pname) {  
        this.pname = pname;  
    }  
}
```

and the code fragment:

```
Product p1 = new Product ("PowerCharger");  
Product p2 = p1;  
System.out.println(p1.equals(p2));  
Product p3 = new Product ("PowerCharger");  
System.out.println(p1.equals(p3));
```

What is the result?

- A. true true  
B. false true  
C. false false  
D. true false

**Answer:** B

#### NEW QUESTION 164

Given the code fragments:

```
class Employee { Optional<Address> address;
Employee (Optional<Address> address) { this.address = address;
}
public Optional<Address> getAddress() { return address; }
}
class Address {
String city = "New York";
public String getCity { return city; } public String toString() {
return city;
}
}
and
Address address = null;
Optional<Address> addrs1 = Optional.ofNullable (address);
Employee e1 = new Employee (addrs1);
String eAddress = (addrs1.isPresent()) ? addrs1.get().getCity() : "City Not available";
What is the result?
```

- A. New York
- B. City Not available
- C. null
- D. A NoSuchElementException is thrown at run time.

**Answer:** B

#### NEW QUESTION 165

Given:

```
class ImageScanner implements AutoCloseable { public void close () throws Exception { System.out.print ("Scanner closed.");
}
public void scanImage () throws Exception { System.out.print ("Scan.");
throw new Exception("Unable to scan.");
}
}
class ImagePrinter implements AutoCloseable { public void close () throws Exception { System.out.print ("Printer closed.");
}
public void printImage () {System.out.print("Print."); }
}
and this code fragment:
try (ImageScanner ir = new ImageScanner(); ImagePrinter iw = new ImagePrinter()) { ir.scanImage();
iw.printImage();
} catch (Exception e) { System.out.print(e.getMessage());
}
What is the result?
```

- A. Scan.Printer close
- B. Scanner close
- C. Unable to scan.
- D. Scan.Scanner close
- E. Unable to scan.
- F. Sca
- G. Unable to scan.
- H. Sca
- I. Unable to sca
- J. Printer closed.

**Answer:** A

#### NEW QUESTION 167

Given:

```
class DataConverter {
    public void copyFlatFilesToTables() { }
    public void close() throws Exception {
        throw new RuntimeException(); // line n1
    }
}
```

and the code fragment:

```
public static void main(String[] args) throws Exception {  
    try (DataConverter dc = new DataConverter()) // line n2  
    { dc.copyFlatFilesToTables(); }  
}
```

What is the result?

- A. A compilation error occurs at line n2.
- B. A compilation error occurs because the try block doesn't have a catch or finally block.
- C. A compilation error occurs at line n1.
- D. The program compiles successfully.

**Answer:** B

#### NEW QUESTION 171

Given the code fragments:

```
public static Optional<String> getCountry(String loc) {  
    Optional<String> couName = Optional.empty();  
    if ("Paris".equals(loc))  
        couName = Optional.of("France");  
    else if ("Mumbai".equals(loc))  
        couName = Optional.of("India");  
    return couName;  
}
```

and

```
Optional<String> city1 = getCountry("Paris");  
Optional<String> city2 = getCountry("Las Vegas");  
System.out.println(city1.orElse("Not Found"));  
if (city2.isPresent())  
    city2.ifPresent(x -> System.out.println(x));  
else  
    System.out.println(city2.orElse("Not Found"));
```

What is the result?

- A. FranceOptional[NotFound]
- B. Optional [France] Optional [NotFound]
- C. Optional[France] Not Found
- D. FranceNot Found

**Answer:** D

#### NEW QUESTION 174

.....

## About ExamBible

*[Your Partner of IT Exam](#)*

## Found in 1998

ExamBible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, ExamBible has its unique advantages that other companies could not achieve.

## Our Advances

### \* 99.9% Uptime

All examinations will be up to date.

### \* 24/7 Quality Support

We will provide service round the clock.

### \* 100% Pass Rate

Our guarantee that you will pass the exam.

### \* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

### NEW QUESTION 1

Given:

```
class Sum extends RecursiveAction { //line n1 static final int THRESHOLD_SIZE = 3;
int stIndex, lstIndex; int [ ] data;
public Sum (int [ ]data, int start, int end) { this.data = data;
this stIndex = start; this. lstIndex = end;
}
protected void compute ( ) { int sum = 0;
if (lstIndex – stIndex <= THRESHOLD_SIZE) { for (int i = stIndex; i < lstIndex; i++) {
sum += data [i];
}
System.out.println(sum);
} else {
new Sum (data, stIndex + THRESHOLD_SIZE, lstIndex).fork( ); new Sum (data, stIndex,
Math.min (lstIndex, stIndex + THRESHOLD_SIZE)
).compute ();
}
}
}
```

and the code fragment:

```
ForkJoinPool fjPool = new ForkJoinPool ( ); int data [ ] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
fjPool.invoke (new Sum (data, 0, data.length));
and given that the sum of all integers from 1 to 10 is 55. Which statement is true?
```

- A. The program prints several values that total 55.
- B. The program prints 55.
- C. A compilation error occurs at line n1.
- D. The program prints several values whose sum exceeds 55.

**Answer:** A

### NEW QUESTION 2

Given the code fragments:

```
public class Test {
    List<String> list = null;
    public void printValues() {
        System.out.print (getList());
    }
    public List<String> getList(){ return list; }
    public void setList(List<String> newList){ list = newList; }
}
```

and

```
List<String> li = Arrays.asList("Dog", "Cat", "Mouse");
Test t = new Test();
t.setList(li.stream().collect(Collectors.toList()));
t.getList().forEach(Test::printValues);
```

What is the result?

- A. null
- B. A compilation error occurs.
- C. DogCatMouse
- D. [Dog, Cat, Mouse]

**Answer:** D

### NEW QUESTION 3

Given the code fragment:

```
public class FileThread implements Runnable { String fName;
public FileThread(String fName) { this.fName = fName; } public void run () System.out.println(fName);}
public static void main (String[] args) throws IOException, InterruptedException {
ExecutorService executor = Executors.newCachedThreadPool(); Stream<Path> listOfFiles = Files.walk(Paths.get("Java Projects")); listOfFiles.forEach(line -> {
executor.execute(new FileThread(line.getFileName().toString ()); //
line n1
});
executor.shutdown(); executor.awaitTermination(5, TimeUnit.DAYS); // line n2
}
}
```

The Java Projects directory exists and contains a list of files. What is the result?

- A. The program throws a runtime exception at line n2.

- B. The program prints files names concurrently.
- C. The program prints files names sequentially.
- D. A compilation error occurs at line n1.

**Answer:** B

#### NEW QUESTION 4

Given the code fragment:

```
List<String> codes = Arrays.asList ("DOC", "MPEG", "JPEG"); codes.forEach (c -> System.out.print(c + " "));  
String fmt = codes.stream()  
.filter (s-> s.contains ("PEG"))  
.reduce((s, t) -> s + t).get(); System.out.println("\n" + fmt); What is the result?
```

- A. DOC MPEG JPEG MPEGJPEG
- B. DOC MPEG MPEGJPEG MPEGMPEGJPEG
- C. MPEGJPEG MPEGJPEG
- D. The order of the output is unpredictable.

**Answer:** A

#### NEW QUESTION 5

Which class definition compiles?

```
A. class Vehicle {  
    int id;  
    public void start() {  
        public class Engine {    int eNo = id;    }  
    }  
}  
  
B. class Computer {  
    private Card sCard = new SoundCard();  
    private abstract class Card { }  
    private class SoundCard extends Card { }  
}  
  
C. class Block {  
    int bno;  
    static class Counter {  
        int locator;  
        Counter() { locator = bno; }  
    }  
}  
  
D. class Product {  
    interface Moveable { void move(); }  
    Moveable mProduct = new Moveable() {  
        void move() { }  
    };  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

#### NEW QUESTION 6

Given:

```
public class Foo {
    public void methodB(String s) { System.out.println("Foo " + s ); }
}

public class Bar extends Foo {
    public void methodB(String s) { System.out.println("Bar " + s); }
}

public class Baz extends Bar {
    public void methodB(String s) { System.out.println("Baz " + s); }
}

public class Daze extends Baz{
    private Bar bb = new Bar();
    public void methodB(String s) {
        bb.methodB(s);
        super.methodB(s);
    }
}

public class TestClass {
    public static void main(String[] args) {
        Baz d = new Daze();
        d.methodB("Hello");
    }
}
```

What is the result?

- A. Bar Hello Foo Hello
- B. Bar Hello Baz Hello
- C. Baz Hello
- D. A compilation error occurs in the Daze class.

**Answer: C**

#### NEW QUESTION 7

Given the code fragment:

```
List<String> cs = Arrays.asList("Java", "Java EE", "Java ME");
// line n1
System.out.print(b);
```

Which code fragment, when inserted at line n1, ensures false is printed?

- A. boolean b = cs.stream().findAny().get().equals("Java");
- B. boolean b = cs.stream().anyMatch(w -> w.equals("Java"));
- C. boolean b = cs.stream().findFirst().get().equals("Java");
- D. boolean b = cs.stream().allMatch(w -> w.equals("Java"));

**Answer: C**

#### NEW QUESTION 8

Given the code fragment:

```
List<String> listVal = Arrays.asList("Joe", "Paul", "Alice", "Tom"); System.out.println (
// line n1
);
```

Which code fragment, when inserted at line n1, enables the code to print the count of string elements whose length is greater than three?

- A. listVal.stream().filter(x -> x.length()>3).count()
- B. listVal.stream().map(x -> x.length()>3).count()
- C. listVal.stream().peek(x -> x.length()>3).count().get()
- D. listVal.stream().filter(x -> x.length()>3).mapToInt(x -> x).count()

**Answer: A**

#### NEW QUESTION 9

Given the content:

```
MessagesBundle.properties file:

inquiry = How are you?

MessagesBundle_de_DE.properties file:

inquiry = Wie geht's?
```

and given the code fragment:

```
Locale currentLocale;
// line 1
ResourceBundle messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);
System.out.println(messages.getString("inquiry"));
```

Which two code fragments, when inserted at line 1 independently, enable the code to print “Wie geht’s?”

- A. currentLocale = new Locale (“de”, “DE”);
- B. currentLocale = new Locale.Builder ().setLanguage (“de”).setRegion (“DE”).build ();
- C. currentLocale = Locale.GERMAN;
- D. currentLocale = new Locale(); currentLocale.setLanguage (“de”); currentLocale.setRegion (“DE”);
- E. currentLocale = Locale.getInstance(Locale.GERMAN,Locale.GERMANY);

**Answer: B**

#### NEW QUESTION 10

Given:

```
public class Foo<K, V> {
    private K key;
    private V value;

    public Foo(K key, V value) { this.key = key; this.value = value; }

    public static <T> Foo<T, T> twice(T value) { return new Foo<T, T>(value, value); }

    public K getKey() { return key; }
    public V getValue() { return value; }
}
```

Which option fails?

- A. Foo<String, Integer> mark = new Foo<String, Integer> (“Steve”, 100);
- B. Foo<String, String> pair = Foo.<String>twice (“Hello World!”);
- C. Foo<Object, Object> percentage = new Foo<String, Integer>(“Steve”, 100);
- D. Foo<String, String> grade = new Foo <> (“John”, “A”);

**Answer: A**

#### NEW QUESTION 10

Given:

```
IntStream stream = IntStream.of (1,2,3); IntFunction<Integer> inFu= x -> y -> x*y; //line n1
IntStream newStream = stream.map(inFu.apply(10)); //line n2 newStream.forEach(System.out::print);
```

Which modification enables the code fragment to compile?

- A. Replace line n1 with: IntFunction<UnaryOperator> inFu = x -> y -> x\*y;
- B. Replace line n1 with: IntFunction<IntUnaryOperator> inFu = x -> y -> x\*y;
- C. Replace line n1 with: BiFunction<IntUnaryOperator> inFu = x -> y -> x\*y;
- D. Replace line n2 with: IntStream newStream = stream.map(inFu.applyAsInt (10));

**Answer: B**

#### NEW QUESTION 12

Given:

```
public class Customer { private String fName; private String lName; private static int count;
public customer (String first, String last) {fName = first, lName = last;
++count;}
static { count = 0; }
public static int getCount() {return count; }
}

public class App {
public static void main (String [] args) { Customer c1 = new Customer(“Larry”, “Smith”);
Customer c2 = new Customer(“Pedro”, “Gonzales”); Customer c3 = new Customer(“Penny”, “Jones”); Customer c4 = new Customer(“Lars”, “Svenson”); c4 =
```

```
null;
c3 = c2;
System.out.println (Customer.getCount());
}
}
```

What is the result?

- A. 2
- B. 3
- C. 4
- D. 5

**Answer: D**

#### NEW QUESTION 17

Given:

```
class Bird {
public void fly () { System.out.print("Can fly"); }
}
class Penguin extends Bird {
public void fly () { System.out.print("Cannot fly"); }
}
```

and the code fragment: class Birdie {  
public static void main (String [ ] args) { fly( ( ) -> new Bird ( ));  
fly (Penguin : : new);  
}  
/\* line n1 \*/  
}

Which code fragment, when inserted at line n1, enables the Birdie class to compile?

- A. static void fly (Consumer<Bird> bird) { bird :: fly ();}
- B. static void fly (Consumer<? extends Bird> bird) {bird.accept( ) fly ();}
- C. static void fly (Supplier<Bird> bird) { bird.get( ) fly ();}
- D. static void fly (Supplier<? extends Bird> bird) { LOST

**Answer: C**

#### NEW QUESTION 19

Given the code fragment:

```
public void recDelete (String dirName) throws IOException { File [ ] listOfFiles = new File (dirName) .listFiles();
if (listOfFiles != null && listOfFiles.length >0) {
for (File aFile : listOfFiles) { if (aFile.isDirectory ()) {
recDelete (aFile.getAbsolutePath ());
} else {
if (aFile.getName ().endsWith (".class")) aFile.delete ();
}
}
}
}
```

Assume that Projects contains subdirectories that contain .class files and is passed as an argument to the recDelete () method when it is invoked. What is the result?

- A. The method deletes all the .class files in the Projects directory and its subdirectories.
- B. The method deletes the .class files of the Projects directory only.
- C. The method executes and does not make any changes to the Projects directory.
- D. The method throws an IOException.

**Answer: A**

#### NEW QUESTION 20

What is true about the java.sql.Statement interface?

- A. It provides a session with the database.
- B. It is used to get an instance of a Connection object by using JDBC drivers.
- C. It provides a cursor to fetch the resulting data.
- D. It provides a class for executing SQL statements and returning the results.

**Answer: D**

#### NEW QUESTION 23

Given:

```
1. abstract class Shape {
2. Shape ( ) { System.out.println ("Shape"); }
3. protected void area ( ) { System.out.println ("Shape"); } 4. }
5.
6. class Square extends Shape {
7. int side;
8. Square int side {
9. /* insert code here */
```

```
10. this.side = side;
11. }
12. public void area ( ) { System.out.println ("Square"); }
13. }
14. class Rectangle extends Square {
15. int len, br;
16. Rectangle (int x, int y) {
17. /* insert code here */
18. len = x, br = y;
19. }
20. void area ( ) { System.out.println ("Rectangle"); }
21. }
```

Which two modifications enable the code to compile? (Choose two.)

- A. At line 1, remove abstract
- B. At line 9, insert super ( );
- C. At line 12, remove public
- D. At line 17, insert super (x);
- E. At line 17, insert super (); super.side = x;
- F. At line 20, use public void area ( ) {

**Answer:** DF

#### NEW QUESTION 26

Given the definition of the Emp class: public class Emp

private String eName; private Integer eAge;

Emp(String eN, Integer eA) { this.eName = eN;

this.eAge = eA;

}

public Integer getEAge () {return eAge;} public String getENAME () {return eName;}

}

and code fragment:

List<Emp>li = Arrays.asList(new Emp("Sam", 20), New Emp("John", 60), New Emp ("Jim", 51));

Predicate<Emp> agVal = s -> s.getEAge() > 50; //line n1 li = li.stream().filter(agVal).collect(Collectors.toList());

Stream<String> names = li.stream().map.(Emp::getENAME); //line n2 names.forEach(n -> System.out.print(n + " "));

What is the result?

- A. Sam John Jim
- B. John Jim
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

**Answer:** B

#### NEW QUESTION 27

Given the records from the Employee table:

eid	ename
111	Tom
112	Jerry
113	Donald

and given the code fragment: try {

Connection conn = DriverManager.getConnection (URL, userName, passWord); Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_UPDATABLE);

st.execute("SELECT\*FROM Employee"); ResultSet rs = st.getResultSet();

while (rs.next()) {

if (rs.getInt(1) ==112) { rs.updateString(2, "Jack");

}

}

rs.absolute(2);

System.out.println(rs.getInt(1) + " " + rs.getString(2));

} catch (SQLException ex) { System.out.println("Exception is raised");

}

Assume that:

The required database driver is configured in the classpath.

The appropriate database accessible with the URL, userName, and passWord exists. What is the result?

- A. The Employee table is updated with the row: 112 Jackand the program prints: 112 Jerry
- B. The Employee table is updated with the row: 112 Jackand the program prints: 112 Jack
- C. The Employee table is not updated and the program prints: 112 Jerry
- D. The program prints Exception is raised.

**Answer:** A

#### NEW QUESTION 29

You want to create a singleton class by using the Singleton design pattern. Which two statements enforce the singleton nature of the design? (Choose two.)

- A. Make the class static.

- B. Make the constructor private.
- C. Override equals() and hashCode() methods of the java.lang.Object class.
- D. Use a static reference to point to the single instance.
- E. Implement the Serializable interface.

**Answer:** BD

### NEW QUESTION 33

Given the code fragment:

```
BiFunction<Integer, Double, Integer> val = (t1, t2) -> t1 + t2; //line n1 System.out.println(val.apply(10, 10.5));
```

What is the result?

- A. 20
- B. 20.5
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

**Answer:** C

### NEW QUESTION 37

Given:

```
class FuelNotAvailException extends Exception { } class Vehicle {  
void ride() throws FuelNotAvailException { //line n1 System.out.println("Happy Journey!");  
}  
}
```

```
class SolarVehicle extends Vehicle {  
public void ride () throws Exception { //line n2 super ride ();  
}  
}
```

and the code fragment:

```
public static void main (String[] args) throws FuelNotAvailException, Exception  
{  
Vehicle v = new SolarVehicle (); v.ride();  
}
```

Which modification enables the code fragment to print Happy Journey!?

- A. Replace line n1 with public void ride() throws FuelNotAvailException {
- B. Replace line n1 with protected void ride() throws Exception {
- C. Replace line n2 with void ride() throws Exception {
- D. Replace line n2 with private void ride() throws FuelNotAvailException {

**Answer:** B

### NEW QUESTION 39

Given the code fragments:

```
public class Book implements Comparator<Book> { String name;  
double price; public Book () {}  
public Book(String name, double price) { this.name = name;  
this.price = price;  
}  
public int compare(Book b1, Book b2) { return b1.name.compareTo(b2.name);  
}  
public String toString() { return name + ":" + price;  
}  
}
```

and

```
List<Book>books = Arrays.asList (new Book ("Beginning with Java", 2), new book ("A  
Guide to Java Tour", 3));  
Collections.sort(books, new Book()); System.out.print(books);
```

What is the result?

- A. [A Guide to Java Tour:3.0, Beginning with Java:2.0]
- B. [Beginning with Java:2, A Guide to Java Tour:3]
- C. A compilation error occurs because the Book class does not override the abstract method compareTo().
- D. An Exception is thrown at run time.

**Answer:** A

### NEW QUESTION 43

Given that course.txt is accessible and contains:

Course : : Java

and given the code fragment:

```
public static void main (String[ ] args) { int i;  
char c;  
try (FileInputStream fis = new FileInputStream ("course.txt"); InputStreamReader isr = new InputStreamReader(fis);) { while (isr.ready()) { //line n1  
isr.skip(2);  
i = isr.read (); c = (char) i;  
System.out.print(c);  
}  
}
```

```
} catch (Exception e) { e.printStackTrace();  
}  
}  
}
```

What is the result?

- A. ur :: va
- B. ueJa
- C. The program prints nothing.
- D. A compilation error occurs at line n1.

**Answer: B**

#### NEW QUESTION 44

Given:

```
interface P { public void method1(); }  
  
interface Q extends P { public void method1(); }  
  
interface R extends P { public void method2(); }  
  
interface S { public default void method() { } }  
  
interface T { public void method1(); public void method2(); }  
  
interface U { public void method1(); public abstract void method2(); }
```

Which two interfaces can you use to create lambda expressions? (Choose two.)

- A. T
- B. R
- C. P
- D. S
- E. Q
- F. U

**Answer: AF**

#### NEW QUESTION 46

In 2015, daylight saving time in New York, USA, begins on March 8th at 2:00 AM. As a result, 2:00 AM becomes 3:00 AM.

Given the code fragment:

```
ZoneId zone = ZoneId.of("America/New_York");  
ZonedDateTime dt = ZonedDateTime.of(LocalDate.of(2015, 3, 8), LocalTime.of(1, 0),  
zone);  
ZonedDateTime dt2 = dt.plusHours(2);  
System.out.print(DateTimeFormatter.ofPattern("H:mm - ").format(dt2));  
System.out.println("difference: " + ChronoUnit.HOURS.between(dt, dt2));
```

Which is the result?

- A. 3:00 – difference: 2
- B. 2:00 – difference: 1
- C. 4:00 – difference: 3
- D. 4:00 – difference: 2

**Answer: B**

#### NEW QUESTION 48

Given:

```
interface Interfacel {
    public default void sayHi() {
        System.out.println("Hi Interface-1");
    }
}

interface Interface2 {
    public default void sayHi() {
        System.out.println("Hi Interface-2");
    }
}

public class MyClass implements Interfacel, Interface2 {
    public static void main(String[] args) {
        Interfacel obj = new MyClass();
        obj.sayHi();
    }
    public void sayHi() {
        System.out.println("Hi MyClass");
    }
}
```

What is the result?

- A. Hi Interface-2
- B. A compilation error occurs.
- C. Hi Interface-1
- D. Hi MyClass

**Answer:** D

#### NEW QUESTION 52

Given the code fragment:

Path source = Paths.get ("/data/december/log.txt"); Path destination = Paths.get("/data");

Files.copy (source, destination);

and assuming that the file /data/december/log.txt is accessible and contains: 10-Dec-2014 – Executed successfully

What is the result?

- A. A file with the name log.txt is created in the /data directory and the content of the /data/december/ log.txt file is copied to it.
- B. The program executes successfully and does NOT change the file system.
- C. A FileNotFoundException is thrown at run time.
- D. A FileAlreadyExistsException is thrown at run time.

**Answer:** D

#### NEW QUESTION 53

Given:

```
public interface LengthValidator {
    public boolean checkLength(String str);
}
```

and

```
public class Txt {
    public static void main(String[] args) {
        boolean res = new LengthValidator() {
            public boolean checkLength(String str) {
                return str.length() > 5 && str.length() < 10;
            }
        }.checkLength("Hello");
    }
}
```

Which interface from the java.util.function package should you use to refactor the class Txt?

- A. Consumer

- B. Predicate
- C. Supplier
- D. Function

**Answer:** C

#### NEW QUESTION 55

Given the code fragment: `Stream<List<String>> iStr= Stream.of ( Arrays.asList ("1", "John"), Arrays.asList ("2", null)0; Stream<<String> nInSt = iStr.flatMapToInt ((x) -> x.stream ()); nInSt.forEach (System.out :: print);` What is the result?

- A. 1John2null
- B. 12
- C. A `NullPointerException` is thrown at run time.
- D. A compilation error occurs.

**Answer:** D

#### NEW QUESTION 58

Given the code fragment:  
`List<String> colors = Arrays.asList("red", "green", "yellow"); Predicate<String> test = n -> { System.out.println("Searching..."); return n.contains("red"); };`  
`colors.stream().filter(c -> c.length() > 3).allMatch(test);` What is the result?

- A. Searching...
- B. Searching...Searching...
- C. Searching...Searching... Searching...
- D. A compilation error occurs.

**Answer:** A

#### NEW QUESTION 63

Given:  
`interface Rideable {Car getCar (String name); } class Car { private String name; public Car (String name) { this.name = name; } }`

Which code fragment creates an instance of `Car`?

- A. `Car auto = Car ("MyCar"): : new;`
- B. `Car auto = Car : : new;Car vehicle = auto : : getCar("MyCar");`
- C. `Rideable rider = Car : : new;Car vehicle = rider.getCar("MyCar");`
- D. `Car vehicle = Rideable : : new : : getCar("MyCar");`

**Answer:** C

#### NEW QUESTION 65

Given the code fragment:  
`List<Integer> nums = Arrays.asList (10, 20, 8); System.out.println ( //line n1 );`  
Which code fragment must be inserted at line n1 to enable the code to print the maximum number in the `nums` list?

- A. `nums.stream().max(Comparator.comparing(a -> a)).get()`
- B. `nums.stream().max(Integer : : max).get()`
- C. `nums.stream().max()`
- D. `nums.stream().map(a -> a).max()`

**Answer:** A

#### NEW QUESTION 68

Given the code fragment:

```
List<Integer> prices = Arrays.asList(3, 4, 5);
prices.stream()
    .filter(e -> e > 4)
    .peek(e -> System.out.print("Price " + e))           // line n1
    .map(n -> n - 1)                                     // line n2
    .peek(n -> System.out.println(" New Price " + n));    // line n3
```

Which modification enables the code to print `Price 5 New Price 4`?

- A. Replace line n2 with `.map (n -> System.out.println ("New Price" + n –1))` and remove line n3

- B. Replace line n2 with `.mapToInt (n -> n - 1);`
- C. Replace line n1 with `.forEach (e -> System.out.print ("Price" + e))`
- D. Replace line n3 with `.forEach (n -> System.out.println ("New Price" + n));`

**Answer:** A

#### NEW QUESTION 71

Given:

```
class RateOfInterest {  
    public static void main (String[] args) { int rateOfInterest = 0;  
    String accountType = "LOAN"; switch (accountType) {  
        case "RD"; rateOfInterest = 5; break;  
        case "FD"; rateOfInterest = 10; break;  
        default:  
            assert false: "No interest for this account"; //line n1  
    }  
    System.out.println ("Rate of interest:" + rateOfInterest);  
    }  
}
```

and the command:

`java -ea RateOfInterest` What is the result?

- A. Rate of interest: 0
- B. An AssertionError is thrown.
- C. No interest for this account
- D. A compilation error occurs at line n1.

**Answer:** B

#### NEW QUESTION 76

Given the code fragment:

```
Deque<String> queue = new ArrayDeque<>();  
queue.add("Susan");  
queue.add("Allen");  
queue.add("David");  
System.out.println(queue.pop());  
System.out.println(queue.remove());  
System.out.println(queue);
```

What is the result?

- A. DavidDavid[Susan, Allen]
- B. SusanSusan[Susan, Allen]
- C. SusanAllen [David]
- D. DavidAllen [Susan]
- E. SusanAllen[Susan, David]

**Answer:** C

#### NEW QUESTION 81

Given the code fragment:

```
// Login time:2015-01-12T21:58:18.817Z  
Instant loginTime = Instant.now();  
Thread.sleep(1000);  
  
// Logout time:2015-01-12T21:58:19.880Z  
Instant logoutTime = Instant.now();  
  
loginTime = loginTime.truncatedTo(ChronoUnit.MINUTES); // line n1  
logoutTime = logoutTime.truncatedTo(ChronoUnit.MINUTES);  
  
if (logoutTime.isAfter(loginTime))  
    System.out.println("Logged out at:"+logoutTime);  
else  
    System.out.println("Can't logout");
```

What is the result?

- A. A compilation error occurs at line n1.

- B. Logged out at: 2015-01-12T21:58:19.880Z
- C. Can't logout
- D. Logged out at: 2015-01-12T21:58:00Z

**Answer:** D

#### NEW QUESTION 85

Assume customers.txt is accessible and contains multiple lines. Which code fragment prints the contents of the customers.txt file?

- A. `Stream<String> stream = Files.find (Paths.get ("customers.txt")); stream.forEach((String c) -> System.out.println(c));`
- B. `Stream<Path> stream = Files.find (Paths.get ("customers.txt")); stream.forEach( c) -> System.out.println(c));`
- C. `Stream<Path> stream = Files.list (Paths.get ("customers.txt")); stream.forEach( c) -> System.out.println(c));`
- D. `Stream<String> lines = Files.lines (Paths.get ("customers.txt")); lines.forEach( c) -> System.out.println(c));`

**Answer:** A

#### NEW QUESTION 86

Given the code fragment:

```
Map<Integer, Integer> mVal = new HashMap<>();
mVal.put(1, 10);
mVal.put(2, 20);
//line n1
c.accept(1, 2);
mVal.forEach(c);
```

Which statement can be inserted into line n1 to print 1,2; 1,10; 2,20;?

- A. `BiConsumer<Integer,Integer> c = (i, j) -> {System.out.print (i + "," + j+ "; ");}`
- B. `BiFunction<Integer, Integer, String> c = (i, j) -> {System.out.print (i + "," + j+ "; ");}`
- C. `BiConsumer<Integer, Integer, String> c = (i, j) -> {System.out.print (i + "," + j+ "; ");}`
- D. `BiConsumer<Integer, Integer, Integer> c = (i, j) -> {System.out.print (i + ","+ j+ "; ");}`

**Answer:** B

#### NEW QUESTION 88

Given:

```
class Vehicle { int vno;
String name;
public Vehicle (int vno, String name) { this.vno = vno;;
this.name = name;
}
public String toString () { return vno + ":" + name;
}
}
```

and this code fragment:

```
Set<Vehicle> vehicles = new TreeSet <> (); vehicles.add(new Vehicle (10123, "Ford")); vehicles.add(new Vehicle (10124, "BMW")); System.out.println(vehicles);
```

What is the result?

- A. 10123 Ford10124 BMW
- B. 10124 BMW10123 Ford
- C. A compilation error occurs.
- D. A ClassCastException is thrown at run time.

**Answer:** D

#### NEW QUESTION 91

Given the code fragment:

```
Map<Integer, String> books = new TreeMap<>(); books.put (1007, "A");
books.put (1002, "C");
books.put (1001, "B");
books.put (1003, "B"); System.out.println (books); What is the result?
```

- A. {1007 = A, 1002 = C, 1001 = B, 1003 = B}
- B. {1001 = B, 1002 = C, 1003 = B, 1007 = A}
- C. {1002 = C, 1003 = B, 1007 = A}
- D. {1007 = A, 1001 = B, 1003 = B, 1002 = C}

**Answer:** B

#### NEW QUESTION 96

Given the records from the STUDENT table:

sid	sname	semail
111	James	james@uni.com
112	Jane	jane@uni.com
114	John	john@uni.com

Given the code fragment:

```
public static void main(String[] args) throws SQLException {
    //code to load and register valid jdbc driver go here
    Connection con = DriverManager.getConnection(URL, username, password);
    Statement st = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
                                       ResultSet.CONCUR_UPDATABLE);

    st.execute("SELECT * FROM student");
    ResultSet rs = st.getResultSet();
    rs.absolute(3);
    rs.moveToInsertRow();
    rs.updateInt(1, 113);
    rs.updateString(2, "Jannet");
    rs.updateString(3, "jannet@uni.com");
    rs.updateRow();
    rs.refreshRow();
    System.out.println(rs.getInt(1) + " : " + rs.getString(2) + " : " + rs.getString
(3));
}
```

Assume that the URL, username, and password are valid. What is the result?

- A. The STUDENT table is not updated and the program prints: 114 : John : john@uni.com
- B. The STUDENT table is updated with the record: 113 : Jannet : jannet@uni.com and the program prints: 114 : John : john@uni.com
- C. The STUDENT table is updated with the record: 113 : Jannet : jannet@uni.com and the program prints: 113 : Jannet : jannet@uni.com
- D. A SQLException is thrown at run time.

**Answer:** A

#### NEW QUESTION 101

Given the code fragments:

```
public class Video {
    public void play() throws IOException {
        System.out.print("Video played.");
    }
}

public class Game extends Video {
    public void play() throws Exception {
        super.play();
        System.out.print("Game played.");
    }
}
```

and

```
try {
    new Game().play();
} catch (Exception e) {
    System.out.print(e.getClass());
}
```

What is the result?

- A. Video played.Game played.
- B. A compilation error occurs.
- C. class java.lang.Exception
- D. class java.io.IOException

Answer: C

#### NEW QUESTION 106

Given:

```
class Engine {  
    double fuelLevel;  
    Engine(int fuelLevel) { this.fuelLevel = fuelLevel; }  
    public void start() {  
        // line n1  
        System.out.println("Started");  
    }  
    public void stop() { System.out.println("Stopped"); }  
}
```

Your design requires that:

- ▶ fuelLevel of Engine must be greater than zero when the start() method is invoked.
- ▶ The code must terminate if fuelLevel of Engine is less than or equal to zero.

Which code fragment should be added at line n1 to express this invariant condition?

- A. `assert (fuelLevel) : "Terminating...";`
- B. `assert (fuelLevel > 0) : System.out.println ("Impossible fuel");`
- C. `assert fuelLevel < 0: System.exit(0);`
- D. `assert fuelLevel > 0: "Impossible fuel" ;`

Answer: C

#### NEW QUESTION 110

Given the code fragment:

```
List<Integer> list1 = Arrays.asList(10, 20); List<Integer> list2 = Arrays.asList(15, 30);  
//line n1
```

Which code fragment, when inserted at line n1, prints 10 20 15 30?

- A. `Stream.of(list1, list2).flatMap(list -> list.stream()).forEach(s -> System.out.print(s + " "));`
- B. `Stream.of(list1, list2).flatMap(list -> list.intStream()).forEach(s -> System.out.print(s + " "));`
- C. `list1.stream().flatMap(list2.stream()).flatMap(e1 -> e1.stream()).forEach(s -> System.out.println(s + " "));`
- D. `Stream.of(list1, list2).flatMapToInt(list -> list.stream()).forEach(s -> System.out.print(s + " "));`

Answer: A

#### NEW QUESTION 115

For which three objects must a vendor provide implementations in its JDBC driver? (Choose three.)

- A. Time
- B. Date
- C. Statement
- D. ResultSet
- E. Connection
- F. SQLException
- G. DriverManager

Answer: CDE

#### Explanation:

Database vendors support JDBC through the JDBC driver interface or through the ODBC connection. Each driver must provide implementations of `java.sql.Connection`, `java.sql.Statement`, `java.sql.PreparedStatement`, `java.sql.CallableStatement`, and `java.sql.ResultSet`. They must also implement the `java.sql.Driver` interface for use by the generic `java.sql.DriverManager` interface.

#### NEW QUESTION 120

Given the code fragment:

```
List<String> qwords = Arrays.asList("why ", "what ", "when ");  
BinaryOperator<String> operator = (s1, s2) -> s1.concat(s2); // line n1  
String sen = qwords.stream()  
    .reduce("Word: ", operator);  
System.out.println(sen);
```

What is the result?

- A. Word: why what when
- B. Word: why Word: why what Word: why what when
- C. Word: why Word: what Word: when
- D. Compilation fails at line n1.

**Answer:** A

#### NEW QUESTION 125

Given the code fragment:

```
9. Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
10. String query = "SELECT id FROM Employee";
11. try (Statement stmt = conn.createStatement()) {
12.     ResultSet rs = stmt.executeQuery(query);
13.     stmt.executeQuery("SELECT id FROM Customer");
14.     while (rs.next()) {
15.         //process the results
16.         System.out.println("Employee ID: "+ rs.getInt("id"));
17.     }
18. } catch (Exception e) {
19.     System.out.println ("Error");
20. }
```

Assume that:

The required database driver is configured in the classpath.

The appropriate database is accessible with the dbURL, userName, and passWord exists.

The Employee and Customer tables are available and each table has id column with a few records and the SQL queries are valid.

What is the result of compiling and executing this code fragment?

- A. The program prints employee IDs.
- B. The program prints customer IDs.
- C. The program prints Error.
- D. compilation fails on line 13.

**Answer:** C

#### NEW QUESTION 130

Given the code fragment:

```
List<String> empDetails = Arrays.asList("100, Robin, HR", "200, Mary, AdminServices",
"101, Peter, HR");
empDetails.stream()
    .filter(s-> s.contains("1"))
    .sorted()
    .forEach(System.out::println); //line n1
```

What is the result?

- A. 100, Robin, HR101, Peter, HR
- B. A compilation error occurs at line n1.
- C. 100, Robin, HR101, Peter, HR200, Mary, AdminServices
- D. 100, Robin, HR200, Mary, AdminServices101, Peter, HR

**Answer:** A

#### NEW QUESTION 131

Given the code fragment:

```
List<String> li = Arrays.asList("Java", "J2EE", "J2ME", "JSTL", "JSP", "Oracle DB");
Predicate<String> val = p -> p.contains("J");
List<String> neLi = li.stream().filter(x -> x.length() > 3)
    .filter(val).collect(Collectors.toList());
System.out.println(neLi);
```

What is the result?

- A. A compilation error occurs.
- B. [Java, J2EE, J2ME, JSTL, JSP]
- C. null
- D. [Java, J2EE, J2ME, JSTL]

**Answer:** A

#### NEW QUESTION 136

Given the code fragment:

```
List<String> nums = Arrays.asList("EE", "SE");
String ans = nums
    .parallelStream()
    .reduce("Java ", (a, b) -> a.concat(b));
System.out.print(ans);
```

What is the result?

- A. Java EEJava EESE
- B. Java EESE

- C. The program prints either:Java EEJava SE orJava SEJava EE  
D. Java EEJava SE

**Answer:** D

#### NEW QUESTION 140

Given the code fragments:

```
class Caller implements Callable<String> { String str;
public Caller (String s) {this.str=s;}
public String call()throws Exception { return str.concat ("Caller");}
}
class Runner implements Runnable { String str;
public Runner (String s) {this.str=s;}
public void run () { System.out.println (str.concat ("Runner"));}
}
and
public static void main (String[] args) InterruptedException, ExecutionException
{
ExecutorService es = Executors.newFixedThreadPool(2); Future f1 = es.submit (new Caller ("Call"));
Future f2 = es.submit (new Runner ("Run")); String str1 = (String) f1.get();
String str2 = (String) f2.get(); //line n1 System.out.println(str1+ ":" + str2);
}
```

What is the result?

- A. The program prints: Run RunnerCall Caller : nullAnd the program does not terminate.  
B. The program terminates after printing: Run RunnerCall Caller : Run  
C. A compilation error occurs at line n1.  
D. An Execution is thrown at run time.

**Answer:** A

#### NEW QUESTION 145

Given:

```
class UserException extends Exception { }
class AgeOutOfLimitException extends UserException { } and the code fragment:
class App {
public void doRegister(String name, int age) throws UserException, AgeOutOfLimitException { if (name.length () < 6) {
throw new UserException ();
} else if (age >= 60) {
throw new AgeOutOfLimitException ();
} else {
System.out.println("User is registered.");
}
}
}
public static void main(String[] args) throws UserException { App t = new App ();
```

- A. t.doRegister("Mathew", 60);}}What is the result?  
B. User is registered.  
C. An AgeOutOfLimitException is thrown.  
D. A UserException is thrown.  
E. A compilation error occurs in the main method.

**Answer:** B

#### NEW QUESTION 148

Which statement is true about the single abstract method of the java.util.function.Function interface?

- A. It accepts one argument and returns void.  
B. It accepts one argument and returns boolean.  
C. It accepts one argument and always produces a result of the same type as the argument.  
D. It accepts an argument and produces a result of any data type.

**Answer:** D

#### NEW QUESTION 153

Given the code fragment:

```
List<String> str = Arrays.asList ("my", "pen", "is", "your", "pen"); Predicate<String> test = s -> {
int i = 0;
boolean result = s.contains ("pen");
System.out.print(i++) + ":"; return result;
};
str.stream()
.filter(test)
.findFirst()
.ifPresent(System.out ::print); What is the result?
```

- A. 0 : 0 : pen  
B. 0 : 1 : pen  
C. 0 : 0 : 0 : 0 : 0 : pen

- D. 0 : 1 : 2 : 3 : 4 :  
E. A compilation error occurs.

**Answer:** A

#### NEW QUESTION 157

Given:

```
public class Emp { String fName; String lName;  
public Emp (String fn, String ln) { fName = fn;  
lName = ln;  
}  
public String getfName() { return fName; } public String getlName() { return lName; }  
}
```

and the code fragment: List<Emp> emp = Arrays.asList ( new Emp ("John", "Smith"),  
new Emp ("Peter", "Sam"),  
new Emp ("Thomas", "Wale")); emp.stream()  
//line n1

.collect(Collectors.toList());

Which code fragment, when inserted at line n1, sorts the employees list in descending order of fName and then ascending order of lName?

- A. .sorted (Comparator.comparing(Emp::getfName).reversed().thenComparing(Emp::getlName))  
B. .sorted (Comparator.comparing(Emp::getfName).thenComparing(Emp::getlName))  
C. .map(Emp::getfName).sorted(Comparator.reverseOrder())  
D. .map(Emp::getfName).sorted(Comparator.reverseOrder()).map (Emp::getlName).reversed

**Answer:** A

#### NEW QUESTION 159

Given:

```
public class Test<T> { private T t;  
public T get () { return t;  
}  
public void set (T t) { this.t = t;  
}  
public static void main (String args [ ] ) { Test<String> type = new Test<>();  
Test type 1 = new Test (); //line n1 type.set("Java");  
type1.set(100); //line n2 System.out.print(type.get() + " " + type1.get());  
}  
}
```

What is the result?

- A. Java 100  
B. java.lang.string@<hashcode>java.lang.Integer@<hashcode>  
C. A compilation error occur  
D. To rectify it, replace line n1 with: Test<Integer> type1 = new Test<>();  
E. A compilation error occur  
F. To rectify it, replace line n2 with: type1.set (Integer(100));

**Answer:** A

#### NEW QUESTION 163

Given:

```
class Product {  
    String pname;  
    public Product(String pname) {  
        this.pname = pname;  
    }  
}
```

and the code fragment:

```
Product p1 = new Product ("PowerCharger");  
Product p2 = p1;  
System.out.println(p1.equals(p2));  
Product p3 = new Product ("PowerCharger");  
System.out.println(p1.equals(p3));
```

What is the result?

- A. true true  
B. false true  
C. false false  
D. true false

**Answer:** B

#### NEW QUESTION 164

Given the code fragments:

```
class Employee { Optional<Address> address;
Employee (Optional<Address> address) { this.address = address;
}
public Optional<Address> getAddress() { return address; }
}
class Address {
String city = "New York";
public String getCity { return city; } public String toString() {
return city;
}
}
and
Address address = null;
Optional<Address> addrs1 = Optional.ofNullable (address);
Employee e1 = new Employee (addrs1);
String eAddress = (addrs1.isPresent()) ? addrs1.get().getCity() : "City Not available";
What is the result?
```

- A. New York
- B. City Not available
- C. null
- D. A NoSuchElementException is thrown at run time.

**Answer:** B

#### NEW QUESTION 165

Given:

```
class ImageScanner implements AutoCloseable { public void close () throws Exception { System.out.print ("Scanner closed.");
}
public void scanImage () throws Exception { System.out.print ("Scan.");
throw new Exception("Unable to scan.");
}
}
class ImagePrinter implements AutoCloseable { public void close () throws Exception { System.out.print ("Printer closed.");
}
public void printImage () {System.out.print("Print."); }
}
and this code fragment:
try (ImageScanner ir = new ImageScanner(); ImagePrinter iw = new ImagePrinter()) { ir.scanImage();
iw.printImage();
} catch (Exception e) { System.out.print(e.getMessage());
}
What is the result?
```

- A. Scan.Printer close
- B. Scanner close
- C. Unable to scan.
- D. Scan.Scanner close
- E. Unable to scan.
- F. Sca
- G. Unable to scan.
- H. Sca
- I. Unable to sca
- J. Printer closed.

**Answer:** A

#### NEW QUESTION 167

Given:

```
class DataConverter {
    public void copyFlatFilesToTables() { }
    public void close() throws Exception {
        throw new RuntimeException(); // line n1
    }
}
```

and the code fragment:

```
public static void main(String[] args) throws Exception {  
    try (DataConverter dc = new DataConverter()) // line n2  
    { dc.copyFlatFilesToTables(); }  
}
```

What is the result?

- A. A compilation error occurs at line n2.
- B. A compilation error occurs because the try block doesn't have a catch or finally block.
- C. A compilation error occurs at line n1.
- D. The program compiles successfully.

**Answer:** B

#### NEW QUESTION 171

Given the code fragments:

```
public static Optional<String> getCountry(String loc) {  
    Optional<String> couName = Optional.empty();  
    if ("Paris".equals(loc))  
        couName = Optional.of("France");  
    else if ("Mumbai".equals(loc))  
        couName = Optional.of("India");  
    return couName;  
}
```

and

```
Optional<String> city1 = getCountry("Paris");  
Optional<String> city2 = getCountry("Las Vegas");  
System.out.println(city1.orElse("Not Found"));  
if (city2.isPresent())  
    city2.ifPresent(x -> System.out.println(x));  
else  
    System.out.println(city2.orElse("Not Found"));
```

What is the result?

- A. FranceOptional[NotFound]
- B. Optional [France] Optional [NotFound]
- C. Optional[France] Not Found
- D. FranceNot Found

**Answer:** D

#### NEW QUESTION 174

.....

## Relate Links

**100% Pass Your 1Z0-809 Exam with Exambible Prep Materials**

<https://www.exambible.com/1Z0-809-exam/>

## Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - <https://www.exambible.com/>