

# 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**

Which code fragment is required to load a JDBC 3.0 driver?

- A. Connection con = Connection.getDriver ("jdbc:xyzdata://localhost:3306/EmployeeDB");
- B. Class.forName("org.xyzdata.jdbc.NetworkDriver");
- C. Connection con = DriverManager.getConnection ("jdbc:xyzdata://localhost:3306/EmployeeDB");
- D. DriverManager.loadDriver ("org.xyzdata.jdbc.NetworkDriver");

**Answer:** B

**NEW QUESTION 4**

Given the code fragment:

```
for (Course a : Course.values()) {  
    System.out.print(a + " Fees " + a.getCost()+" " );  
}
```

Which is the valid definition of the Course enum?

A. 

```
enum Course { JAVA(100), J2ME(150);  
    private int cost;  
    public Course(int c) {  
        this.cost = c;  
    }  
    int getCost() {  
        return cost;  
    }  
}
```

B. 

```
enum Course { JAVA(100), J2ME(150);  
    private static int cost;  
    private Course(int c) {  
        this.cost = c;  
    }  
    static int getCost() {  
        return cost;  
    }  
}
```

C. 

```
final enum Course { JAVA(100), J2ME(150);  
    private int cost;  
    public Course(int c) {  
        this.cost = c;  
    }  
    int getCost() {  
        return cost;  
    }  
    void setCost(int c) {  
        this.cost = c;  
    }  
}
```

D. 

```
enum Course { JAVA(100), J2ME(150);  
    private int cost;  
    Course(int c) {  
        this.cost = c;  
    }  
    int getCost() {  
        return cost;  
    }  
}
```

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

**Answer:** A

**NEW QUESTION 5**

Which two statements are true about the Fork/Join Framework? (Choose two.)

- A. The RecursiveTask subclass is used when a task does not need to return a result.
- B. The Fork/Join framework can help you take advantage of multicore hardware.
- C. The Fork/Join framework implements a work-stealing algorithm.
- D. The Fork/Join solution when run on multicore hardware always performs faster than standard sequential solution.

**Answer:** AC

**NEW QUESTION 6**

Given:

```
class Resource implements AutoCloseable {  
    public void close() throws Exception {  
        System.out.print("Close-");  
    }  
    public void open() {  
        System.out.print("Open-");  
    }  
}
```

and this code fragment:

```
Resource res1 = new Resource();  
try {  
    res1.open();  
    res1.close();  
} catch (Exception e) {  
    System.out.println("Exception - 1");  
}  
try (res1 = new Resource()) { // line n1  
    res1.open();  
} catch (Exception e) {  
    System.out.println("Exception - 2");  
}
```

What is the result?

- A. Open-Close- Exception – 1 Open-Close-
- B. Open-Close-Open-Close-
- C. A compilation error occurs at line n1.
- D. Open-Close-Open-

**Answer:** C

**NEW QUESTION 7**

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 8**

Given the code fragment:

```
List<Integer> values = Arrays.asList (1, 2, 3); values.stream ()  
.map(n -> n*2) //line n1  
.peek(System.out::print) //line n2  
.count();
```

What is the result?

- A. 246

- B. The code produces no output.
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

**Answer:** A

#### NEW QUESTION 9

Given the code fragment:

```
ProductCode<Number, Integer> c1 = new ProductCode<Number, Integer>(); /* c1
instantiation */
ProductCode<Number, String> c2 = new ProductCode<Number, String>();    /* c2
instantiation */
```

You have been asked to define the ProductCode class. The definition of the ProductCode class must allow c1 instantiation to succeed and cause a compilation error on c2 instantiation.

Which definition of ProductCode meets the requirement?

```
A. class ProductCode<T, S<Integer>> {
    T c1;
    S c2;
}

B. class ProductCode<T, S extends T> {
    T c1;
    S c2;
}

C. class ProductCode<T, S> {
    T c1;
    S c2;
}

D. class ProductCode<T, S super T> {
    T c1;
    S c2;
}
```

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

**Answer:** B

#### NEW QUESTION 10

Given the code fragment:

Path file = Paths.get ("courses.txt");

// line n1

Assume the courses.txt is accessible.

Which code fragment can be inserted at line n1 to enable the code to print the content of the courses.txt file?

- A. List<String> fc = Files.list(file); fc.stream().forEach (s -> System.out.println(s));
- B. Stream<String> fc = Files.readAllLines (file); fc.forEach (s -> System.out.println(s));
- C. List<String> fc = readAllLines(file); fc.stream().forEach (s -> System.out.println(s));
- D. Stream<String> fc = Files.lines (file); fc.forEach (s -> System.out.println(s));

**Answer:** D

#### NEW QUESTION 10

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 12

Given the code fragment:

```
List<String> valList = Arrays.asList("", "George", "", "John", "Jim");  
Long newVal = valList.stream()           // line n1  
    .filter(x -> !x.isEmpty())  
    .count();                             // line n2  
System.out.print(newVal);
```

What is the result?

- A. A compilation error occurs at line n2.
- B. 3
- C. 2
- D. A compilation error occurs at line n1.

**Answer:** A

#### NEW QUESTION 13

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 17

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 22

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 24

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 25

Given:

```
public class Counter {  
public static void main (String[ ] args) { int a = 10;  
int b = -1;  
assert (b >=1) : "Invalid Denominator"; int = a / b;  
System.out.println (c);  
}  
}
```

What is the result of running the code with the `-ea` option?

- A. -10
- B. An AssertionError is thrown.
- C. A compilation error occurs.

**Answer: C**

#### NEW QUESTION 27

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 28

Given the content of `/resources/Message.properties`: `welcome1="Good day!"`

and given the code fragment: `Properties prop = new Properties ();`

```
FileInputStream fis = new FileInputStream ("/resources/Message.properties"); prop.load(fis);
```

```
System.out.println(prop.getProperty("welcome1")); System.out.println(prop.getProperty("welcome2", "Test")); //line n1
```

```
System.out.println(prop.getProperty("welcome3"));
```

What is the result?

- A. Good day!Testfollowed by an Exception stack trace
- B. Good day!followed by an Exception stack trace
- C. Good day!Test null
- D. A compilation error occurs at line n1.

**Answer: C**

#### NEW QUESTION 30

Given the code fragment: `public class Foo {`

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

```
Map<Integer, String> unsortMap = new HashMap< > ( ); unsortMap.put (10, "z");
```

```
unsortMap.put (5, "b");
```

```
unsortMap.put (1, "d");
```

```
unsortMap.put (7, "e");
```

```
unsortMap.put (50, "j");
```

```
Map<Integer, String> treeMap = new TreeMap <Integer, String> (new Comparator<Integer> ( ) {
```

```
@Override public int compare (Integer o1, Integer o2) {return o2.compareTo
```

```
(o1); } } );
```

```
treeMap.putAll (unsortMap);
```

```
for (Map.Entry<Integer, String> entry : treeMap.entrySet ( ) ) { System.out.print (entry.getValue ( ) + " ");
```

```
}
```

```
}
```

```
}  
What is the result?
```

- A. A compilation error occurs.
- B. d b e z j
- C. j z e b d
- D. z b d e j

**Answer: C**

#### NEW QUESTION 33

Given the code fragments:

```
class MyThread implements Runnable {  
    private static AtomicInteger count = new AtomicInteger (0); public void run () {  
        int x = count.incrementAndGet(); System.out.print (x+" ");  
    }  
}
```

and

```
Thread thread1 = new Thread(new MyThread()); Thread thread2 = new Thread(new MyThread()); Thread thread3 = new Thread(new MyThread()); Thread [] ta =  
{thread1, thread2, thread3};  
for (int x= 0; x < 3; x++) { ta[x].start();  
}
```

Which statement is true?

- A. The program prints 1 2 3 and the order is unpredictable.
- B. The program prints 1 2 3.
- C. The program prints 1 1 1.
- D. A compilation error occurs.

**Answer: A**

#### NEW QUESTION 34

Given that /green.txt and /colors/yellow.txt are accessible, and the code fragment: Path source = Paths.get("/green.txt");

Path target = Paths.get("/colors/yellow.txt");

Files.move(source, target, StandardCopyOption.ATOMIC\_MOVE); Files.delete(source);

Which statement is true?

- A. The green.txt file content is replaced by the yellow.txt file content and the yellow.txt file is deleted.
- B. The yellow.txt file content is replaced by the green.txt file content and an exception is thrown.
- C. The file green.txt is moved to the /colors directory.
- D. A FileAlreadyExistsException is thrown at runtime.

**Answer: D**

#### NEW QUESTION 39

Given the code fragments:

```
4. void doStuff() throws ArithmeticException, NumberFormatException, Exception
```

```
{
```

```
5. if (Math.random() >= 1) throw new Exception ("Try again"); 6. }
```

```
and
```

```
24. try {
```

```
25. doStuff ( );
```

```
26. } catch (ArithmeticException | NumberFormatException | Exception e) {
```

```
27. System.out.println (e.getMessage()); }
```

```
28. catch (Exception e) {
```

```
29. System.out.println (e.getMessage()); }
```

```
30. }
```

Which modification enables the code to print Try again?

- A. Comment the lines 28, 29 and 30.
- B. Replace line 26 with: } catch (Exception | ArithmeticException | NumberFormatException e) {
- C. Replace line 26 with: } catch (ArithmeticException | NumberFormatException e) {
- D. Replace line 27 with: throw e;

**Answer: C**

#### NEW QUESTION 44

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 47

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 50

Given that data.txt and alldata.txt are accessible, and the code fragment:

```
public void writeFiles() throws IOException {
    BufferedReader br = new BufferedReader(new FileReader("data.txt"));
    BufferedWriter bw = new BufferedWriter(new FileWriter("alldata.txt"));
    String line = null;
    while ((line = br.readLine()) != null) {
        bw.append(line + "\n");
    }
    // line n1
}
```

What is required at line n1 to enable the code to overwrite alldata.txt with data.txt?

- A. br.close();
- B. bw.writeIn();
- C. br.flush();
- D. bw.flush();

**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 54

Given:

```
public enum USCurrency { PENNY (1),
    NICKLE(5), DIME (10), QUARTER(25);
    private int value;
    public USCurrency(int value) { this.value = value;
    }
    public int getValue() {return value;}
}
public class Coin {
    public static void main (String[] args) { USCurrency usCoin =new USCurrency.DIME; System.out.println(usCoin.getValue());
    }
}
```

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

- A. Nest the USCurrency enumeration declaration within the Coin class.
- B. Make the USCurrency enumeration constructor private.
- C. Remove the new keyword from the instantiation of usCoin.
- D. Make the getter method of value as a static method.
- E. Add the final keyword in the declaration of value.

**Answer:** BC

#### NEW QUESTION 57

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 62

Given the code fragment:

```
//line n1
Double d = str.average().getAsDouble();
System.out.println("Average = " + d);
```

Which should be inserted into line n1 to print Average = 2.5?

- A. IntStream str = Stream.of (1, 2, 3, 4);
- B. IntStream str = IntStream.of (1, 2, 3, 4);
- C. DoubleStream str = Stream.of (1.0, 2.0, 3.0, 4.0);
- D. Stream str = Stream.of (1, 2, 3, 4);

**Answer:** C

**NEW QUESTION 65**

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 68**

Given:

Item table

- ID, INTEGER: PK
- DESCRIP, VARCHAR(100)
- PRICE, REAL
- QUANTITY< INTEGER

And given the code fragment:

```
9. try {
10. Connection conn = DriverManager.getConnection(dbURL, username, password);
11. String query = "Select * FROM Item WHERE ID = 110";
12. Statement stmt = conn.createStatement();
13. ResultSet rs = stmt.executeQuery(query);
14. while(rs.next()) {
15. System.out.println("ID: " + rs.getInt("Id"));
16. System.out.println("Description: " + rs.getString("Descrip"));
17. System.out.println("Price: " + rs.getDouble("Price"));
18. System.out.println("Quantity: " + rs.getInt("Quantity"));
19. }
20. } catch (SQLException se) {
21. System.out.println("Error");
22. }
```

Assume that:

The required database driver is configured in the classpath.

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

What is the result?

- A. An exception is thrown at runtime.
- B. Compilation fails.
- C. The code prints Error.
- D. The code prints information about Item 110.

**Answer: D**

**NEW QUESTION 69**

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 73**

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 74**

Given:

```
class Block {
    String color;
    int size;
    Block(int size, String color) {
        this.size = size;
        this.color = color;
    }
}
```

and the code fragment:

```
List<Block> blocks = new ArrayList<>();
blocks.add(new Block(10, "Green"));
blocks.add(new Block(7, "Red"));
blocks.add(new Block(12, "Blue"));
Collections.sort(blocks, new ColorSorter());
```

Which definition of the ColorSorter class sorts the blocks list?



```
A. class ColorSorter implements Comparable<Block> {
    public boolean compare(Block o1, Block o2) {
        return o1.color.equals(o2.color);
    }
}

B. class ColorSorter implements Comparable<Block> {
    public int compareTo(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}

C. class ColorSorter implements Comparator<Block> {
    public int compare(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}

D. class ColorSorter implements Comparator<Block> {
    public boolean compare(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}
```

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

**Answer:** B

#### NEW QUESTION 77

The data.doc, data.txt and data.xml files are accessible and contain text. Given the code fragment:

```
Stream<Path> paths = Stream.of (Paths. get("data.doc"),
Paths. get("data.txt"),
Paths. get("data.xml"));
paths.filter(s-> s.toString().endsWith("txt")).forEach( s -> {
try { Files.readAllLines(s)
.stream()
.f orEach(System.out::println); //line n1
} catch (IOException e) { System.out.println("Exception");
}
});
```

What is the result?

- A. The program prints the content of data.txt file.
- B. The program prints: Exception<<The content of the data.txt file>> Exception
- C. A compilation error occurs at line n1.
- D. The program prints the content of the three files.

**Answer:** A

#### NEW QUESTION 80

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 85

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 86

Given the code fragment: UnaryOperator<Integer> uo1 = s -> s\*2; line n1  
List<Double> loanValues = Arrays.asList(1000.0, 2000.0); loanValues.stream()  
.filter(lv -> lv >= 1500)  
.map(lv -> uo1.apply(lv))  
.forEach(s -> System.out.print(s + " ")); What is the result?

- A. 4000.0
- B. 4000
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Answer: D

#### NEW QUESTION 89

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 91

Given:

```
public class product { int id; int price;  
public Product (int id, int price) { this.id = id;  
this.price = price;  
}  
public String toString() { return id + ":" + price; }  
}
```

and the code fragment:

```
List<Product> products = Arrays.asList(new Product(1, 10), new Product (2, 30),  
new Product (2, 30));  
Product p = products.stream().reduce(new Product (4, 0), (p1, p2) -> { p1.price+=p2.price;  
return new Product (p1.id, p1.price);}); products.add(p); products.stream().parallel()  
.reduce((p1, p2) -> p1.price > p2.price ? p1 : p2)  
.ifPresent(System.out::println); What is the result?
```

- A. 2 : 30  
B. 4 : 0  
C. 4 : 60  
D. 4 : 602 : 303 : 201 : 10  
E. The program prints nothing.

**Answer:** C

#### NEW QUESTION 94

Which statement is true about java.util.stream.Stream?

- A. A stream cannot be consumed more than once.  
B. The execution mode of streams can be changed during processing.  
C. Streams are intended to modify the source data.  
D. A parallel stream is always faster than an equivalent sequential stream.

**Answer:** B

#### NEW QUESTION 99

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 104

Given the code fragment:

```
class CallerThread implements Callable<String> { String str;  
public CallerThread(String s) {this.str=s;} public String call() throws Exception { return str.concat("Call");  
}  
}  
and  
public static void main (String[] args) throws InterruptedException, ExecutionException  
{  
ExecutorService es = Executors.newFixedThreadPool(4); //line n1 Future f1 = es.submit (new CallerThread("Call"));  
String str = f1.get().toString(); System.out.println(str);  
}
```

Which statement is true?

- A. The program prints Call Call and terminates.  
B. The program prints Call Call and does not terminate.  
C. A compilation error occurs at line n1.  
D. An ExecutionException is thrown at run time.

**Answer:** B

**NEW QUESTION 106**

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 108**

Given:

```
class MyClass implements AutoCloseable {  
    int test;  
    public void close() { }  
    public MyClass copyObject() { return this; }  
}
```

and the code fragment:

```
MyClass obj = null;  
try (MyClass obj1 = new MyClass()) {  
    obj1.test = 100;  
    obj = obj1.copyObject(); // line n1  
}  
System.out.println(obj.test); // line n2
```

What is the result?

- A. An exception is thrown at line n2.
- B. 100
- C. A compilation error occurs because the try block is declared without a catch or finally block.
- D. A compilation error occurs at line n1.

**Answer:** D

**NEW QUESTION 112**

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 116**

Which two methods from the java.util.stream.Stream interface perform a reduction operation? (Choose two.)

- A. count ()
- B. collect ()
- C. distinct ()
- D. peek ()
- E. filter ()

**Answer:** AB

**NEW QUESTION 118**

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 121

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 124

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 127

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 128

Which action can be used to load a database driver by using JDBC3.0?

- A. Add the driver class to the META-INF/services folder of the JAR file.
- B. Include the JDBC driver class in a jdbc.properties file.
- C. Use the java.lang.Class.forName method to load the driver class.
- D. Use the DriverManager.getDriver method to load the driver class.

**Answer:** C

#### NEW QUESTION 133

.....



## Thank You for Trying Our Product

### We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### 1Z0-809 Practice Exam Features:

- \* 1Z0-809 Questions and Answers Updated Frequently
- \* 1Z0-809 Practice Questions Verified by Expert Senior Certified Staff
- \* 1Z0-809 Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* 1Z0-809 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

**100% Actual & Verified — Instant Download, Please Click**  
**[Order The 1Z0-809 Practice Test Here](#)**