



**Oracle**

## **Exam Questions 1Z0-809**

Java SE 8 Programmer II

### NEW QUESTION 1

Given:

```
class Book { int id;
String name;
public Book (int id, String name) { this.id = id;
this.name = name;
}
public boolean equals (Object obj) { //line n1 boolean output = false;
Book b = (Book) obj;
if (this.name.equals(b.name)) output = true;
}
return output;
}
}
```

and the code fragment:

```
Book b1 = new Book (101, "Java Programing"); Book b2 = new Book (102, "Java Programing"); System.out.println (b1.equals(b2)); //line n2 Which statement is true?
```

- A. The program prints true.
- B. The program prints false.
- C. A compilation error occur
- D. To ensure successful compilation, replace line n1 with: boolean equals (Book obj) {
- E. A compilation error occur
- F. To ensure successful compilation, replace line n2 with: System.out.println (b1.equals((Object) b2));

**Answer:** A

### NEW QUESTION 2

Given the code fragment:

```
public static void main (String[] args) throws IOException { BufferedReader brCopy = null;
try (BufferedReader br = new BufferedReader (new FileReader("employee.txt")))
{ // line n1
br.lines().forEach(c -> System.out.println(c)); brCopy = br; //line n2
}
brCopy.ready(); //line n3;
}
```

Assume that the ready method of the BufferedReader, when called on a closed BufferedReader, throws an exception, and employee.txt is accessible and contains valid text.

What is the result?

- A. A compilation error occurs at line n3.
- B. A compilation error occurs at line n1.
- C. A compilation error occurs at line n2.
- D. The code prints the content of the employee.txt file and throws an exception at line n3.

**Answer:** D

### NEW QUESTION 3

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 4**

What is the result?

```
7. BiPredicate<String, String> bp = (String s1, String s2) -> s1.contains("SG") &&  
   s2.contains("Java");  
8. BiFunction<String, String, Integer> bf = (String s1, String s2) -> {  
9.     int fee = 0;  
10.    if (bp.test(s1, s2)) {  
11.        fee = 100;  
12.    }  
13.    return fee;  
14. };  
15. int fee1 = bf.apply("D101SG", "Java Programming");  
16. System.out.println(fee1);
```

- A. A compilation error occurs at line 7.
- B. 100
- C. A compilation error occurs at line 8.
- D. A compilation error occurs at line 15.

**Answer:** A**NEW QUESTION 5**

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 6

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 7

Which statement is true about java.time.Duration?

- A. It tracks time zones.
- B. It preserves daylight saving time.
- C. It defines time-based values.
- D. It defines date-based values.

**Answer:** C

#### NEW QUESTION 8

Given:

```
public class Product {
    public double applyDiscount(double price) {
        assert (price > 0); // line n1
        return price * 0.50;
    }
    public static void main(String[] args) {
        Product p = new Product();
        double newPrice =
            p.applyDiscount(Double.parseDouble(args[0]));
        System.out.println("New Price: " + newPrice);
    }
}
```

and the command: java Product 0 What is the result?

- A. An AssertionError is thrown.
- B. A compilation error occurs at line n1.
- C. New Price: 0.0
- D. A NumberFormatException is thrown at run time.

**Answer:** D

#### NEW QUESTION 9

Given the code fragment:

```
List<String> words = Arrays.asList("win", "try", "best", "luck", "do");
Predicate<String> test1 = w -> {
    System.out.println("Checking...");
    return w.equals("do"); // line n1
};
Predicate test2 = (String w) -> w.length() > 3; // line n2
words.stream()
    .filter(test2)
    .filter(test1)
    .count();
```

What is the result?

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

**Answer:** A

#### NEW QUESTION 10

Given the code fragment:

```
List<Integer> values = Arrays.asList(1, 2, 3); values.stream ()
.map(n -> n*2) //line n1
.p eek(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 10

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 12

Given:

```
public class Job {
    String name;
    Integer cost;
    Job(String name, Integer cost) {
        this.name = name;
        this.cost = cost;
    }
    String getName() { return name; }
    int getCost() { return cost; }
    public static void main(String[] args) {
        Job j1 = new Job("IT", null);
        DoubleSupplier jS1 = j1::getCost;
        System.out.println(j1.getName() + ":" + jS1.getAsDouble());
    }
}
```

What is the result?

- A. IT:null
- B. A NullPointerException is thrown at run time.
- C. A compilation error occurs.
- D. IT:0.0

**Answer:** D

#### NEW QUESTION 15

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 19

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 22

Given:

```
public final class IceCream { public void prepare() {}
}
```

```
public class Cake {
    public final void bake(int min, int temp) {} public void mix() {}
}
```

```
public class Shop {
    private Cake c = new Cake (); private final double discount = 0.25;
    public void makeReady () { c.bake(10, 120); }
}
```

```
public class Bread extends Cake {
    public void bake(int minutes, int temperature) {} public void addToppings() {}
}
```

Which statement is true?

- A. A compilation error occurs in IceCream.

- B. A compilation error occurs in Cake.
- C. A compilation error occurs in Shop.
- D. A compilation error occurs in Bread
- E. All classes compile successfully.

**Answer:** D

#### NEW QUESTION 27

Which two statements are true about synchronization and locks? (Choose two.)

- A. A thread automatically acquires the intrinsic lock on a synchronized statement when executed.
- B. The intrinsic lock will be retained by a thread if return from a synchronized method is caused by an uncaught exception.
- C. A thread exclusively owns the intrinsic lock of an object between the time it acquires the lock and the time it releases it.
- D. A thread automatically acquires the intrinsic lock on a synchronized method's object when entering that method.
- E. Threads cannot acquire intrinsic locks on classes.

**Answer:** AB

#### NEW QUESTION 31

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 36

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 38

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 42

Given the code fragment:

```
Path path1 = Paths.get("/app/.sys/"); Path res1 = path1.resolve("log");
Path path2 = Paths.get("/server/exe/"); Path res1 = path1.resolve("/readme/"); System.out.println(res1); System.out.println(res2);
What is the result?
```

- A. /app/sys/log/readme/server/exe
- B. /app/log/sys/server/exe/readme
- C. /app/.sys/log/readme
- D. /app/.sys/log/server/exe/readme

**Answer:** C

#### NEW QUESTION 43

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 48

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 52

Locale	Currency Symbol	Currency Code
US	\$	USD

and the code fragment?

```
double d = 15;
Locale l = new Locale("en", "US");
NumberFormat formatter = NumberFormat.getCurrencyInstance(l);
System.out.println(formatter.format(d));
```

What is the result?

- A. \$15.00
- B. 15 \$
- C. USD 15.00
- D. USD \$15

**Answer:** A

#### NEW QUESTION 53

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 54

Which statement is true about the DriverManager class?

- A. It returns an instance of Connection.
- B. it executes SQL statements against the database.
- C. It only queries metadata of the database.
- D. it is written by different vendors for their specific database.

**Answer:** A

#### Explanation:

The DriverManager returns an instance of Doctrine\DBAL\Connection which is a wrapper around the underlying driver connection (which is often a PDO instance).

#### NEW QUESTION 59

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 61

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 62

Given the code fragment:

```
ZonedDateTime depart = ZonedDateTime.of(2015, 1, 15, 3, 0, 0, 0, ZoneId.of("UTC-7"));
ZonedDateTime arrive = ZonedDateTime.of(2015, 1, 15, 9, 0, 0, 0, ZoneId.of("UTC-5"));
long hrs = ChronoUnit.HOURS.between(depart, arrive); //line n1
System.out.println("Travel time is" + hrs + "hours");
```

What is the result?

- A. Travel time is 4 hours
- B. Travel time is 6 hours
- C. Travel time is 8 hours
- D. An exception is thrown at line n1.

**Answer: A**

#### NEW QUESTION 64

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.writeLn();
- C. br.flush();
- D. bw.flush();

**Answer: D**

#### NEW QUESTION 65

Given the code fragment:

```
Stream<List<String>> strs = Stream.of(
    Arrays.asList("text1", "text2"),
    Arrays.asList("text2", "text3"));
Stream<String> bs2 = strs
    .filter(b -> b.contains("text1"))
    .flatMap(rs -> rs.stream());
bs2.forEach(b -> System.out.print(b));
```

What is the result?

- A. text1text2
- B. text1text2text2text3
- C. text1
- D. [text1, text2]

**Answer: A**

**NEW QUESTION 68**

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 72**

Given:

```
public class Canvas implements Drawable { public void draw () { }  
}  
public abstract class Board extends Canvas {  
public class Paper extends Canvas { protected void draw (int color) { }  
}  
public class Frame extends Canvas implements Drawable { public void resize () { }  
}  
public interface Drawable { public abstract void draw ();  
}
```

Which statement is true?

- A. Board does not compile.
- B. Paper does not compile.
- C. Frame does not compile.
- D. Drawable does not compile.
- E. All classes compile successfully.

**Answer: E****NEW QUESTION 74**

Given:

```
class Student {  
String course, name, city;  
public Student (String name, String course, String city) { this.course = course; this.name = name; this.city = city;  
}  
public String toString() {  
return course + ":" + name + ":" + city;  
}
```

and the code fragment: `List<Student> stds = Arrays.asList(  
new Student ("Jessy", "Java ME", "Chicago"), new Student ("Helen", "Java EE", "Houston"), new Student ("Mark", "Java ME", "Chicago")); stds.stream()  
.collect(Collectors.groupingBy(Student::getCourse))  
.forEachOrdered((src, res) -> System.out.println(src));` What is the result?

- A. [Java EE: Helen:Houston][Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
- B. Java EEJava ME
- C. [Java ME: Jessy:Chicago, Java ME: Mark:Chicago] [Java EE: Helen:Houston]
- D. A compilation error occurs.

**Answer: B****NEW QUESTION 79**

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()
.f ilter(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 80

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 83

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 86

Given the code fragment:

```
List<String> nL = Arrays.asList("Jim", "John", "Jeff"); Function<String, String> funVal = s -> "Hello : ".contact(s); nL.Stream()
.map(funVal)
.peek(System.out::print); What is the result?
```

- A. Hello : Jim Hello : John Hello : Jeff
- B. Jim John Jeff
- C. The program prints nothing.
- D. A compilation error occurs.

**Answer:** C

#### NEW QUESTION 88

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 91

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 96

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 97

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 102

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 103

Given:

```
public class StrMan {
    public static void doStuff(String s) {
        try {
            if (s == null) {
                throw new NullPointerException();
            }
        } finally {
            System.out.println("-finally-");
        }
        System.out.println("-doStuff-");
    }
    public static void main (String[] args) {
        try {
            doStuff(null);
        } catch (NullPointerException npe) {
            System.out.println("-catch-");
        }
    }
}
```

What is the result?

- A. -catch--finally--dostuff-
- B. -catch-
- C. -finally--catch-
- D. -finally-dostuff--catch-

**Answer:** C

#### NEW QUESTION 106

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 110

Given:

```
interface Doable {
    public void doSomething (String s);
}
```

Which two class definitions compile? (Choose two.)

- A. public abstract class Task implements Doable { public void doSomethingElse(String s) { }}
- B. public abstract class Work implements Doable { public abstract void doSomething(String s) { } public void doYourThing(Boolean b) { }}
- C. public class Job implements Doable { public void doSomething(Integer i) { }}
- D. public class Action implements Doable { public void doSomething(Integer i) { } public String doThis(Integer i) { }}
- E. public class Do implements Doable { public void doSomething(Integer i) { } public void doSomething(String s) { } public void doThat (String s) { }}

**Answer:** AE

#### NEW QUESTION 112

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 117

Given the code fragments: `class TechName {  
String techName;  
TechName (String techName) { this.techName=techName;  
}  
}`  
and  
`List<TechName> tech = Arrays.asList ( new TechName("Java-"),  
new TechName("Oracle DB-"), new TechName("J2EE-")  
);  
Stream<TechName> stre = tech.stream();  
//line n1`  
Which should be inserted at line n1 to print Java-Oracle DB-J2EE-?

A. `stre.forEach(System.out::print);`  
B. `stre.map(a-> a.techName).forEach(System.out::print);`  
C. `stre.map(a-> a).forEachOrdered(System.out::print);`  
D. `stre.forEachOrdered(System.out::print);`

**Answer:** B

#### NEW QUESTION 118

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 122

Given:

```
class Product {  
    String name;  
    int qty;  
    public String toString(){  
        return name;  
    }  
    public Product(String name, int qty) {  
        this.name = name;  
        this.qty = qty;  
    }  
    static class ProductFilter {  
        public boolean isAvailable(Product p) {    // line n1  
            return p.qty >= 10;  
        }  
    }  
}
```

and the code fragment:

```
List<Product> products = Arrays.asList(  
    new Product("MotherBoard", 5),  
    new Product("Speaker", 20));  
products.stream()  
    .filter(Product.ProductFilter::isAvailable) // line n2  
    .forEach(p -> System.out.println(p));
```

Which modification enables the code fragment to print Speaker?

- A. Implement Predicate in the Product.ProductFilter class and replace line n2 with .filter (p-> p.ProductFilter.test (p))
- B. Replace line n1 with:public static boolean isAvailable (Product p) {
- C. Replace line n2 with:.filter (p -> p.ProductFilter: :isAvailable (p))
- D. Replace line n2 with:.filter (p -> Product: :ProductFilter: :isAvailable ())

**Answer:** B

#### NEW QUESTION 123

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 124

Given:

```
public interface Moveable<Integer> {
public default void walk (Integer distance) {System.out.println("Walking");} public void run(Integer distance);
}
```

Which statement is true?

- A. Moveable can be used as below:Moveable<Integer> animal = n -> System.out.println("Running" + n); animal.run(100);animal.walk(20);
- B. Moveable can be used as below: Moveable<Integer> animal = n -> n + 10; animal.run(100);animal.walk(20);
- C. Moveable can be used as below:Moveable animal = (Integer n) -> System.out.println(n); animal.run(100);Moveable.walk(20);
- D. Movable cannot be used in a lambda expression.

**Answer:** A

#### NEW QUESTION 127

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 130

Given the code fragment:



```
10. try {
11.     Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
12.     String query = "SELECT * FROM Employee WHERE ID = 110";
13.     Statement stmt = conn.createStatement();
14.     ResultSet rs = stmt.executeQuery(query);
15.     System.out.println("Employee ID: " + rs.getInt("ID"));
16. } catch (Exception se) {
17.     System.out.println("Error");
18. }
```

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 table has a column ID of type integer and the SQL query matches one record. What is the result?

- A. Compilation fails at line 14.
- B. Compilation fails at line 15.
- C. The code prints the employee ID.
- D. The code prints Error.

**Answer:** A

#### NEW QUESTION 134

Which two reasons should you use interfaces instead of abstract classes? (Choose two.)

- A. You expect that classes that implement your interfaces have many common methods or fields, or require access modifiers other than public.
- B. You expect that unrelated classes would implement your interfaces.
- C. You want to share code among several closely related classes.
- D. You want to declare non-static on non-final fields.
- E. You want to take advantage of multiple inheritance of type.

**Answer:** BE

#### NEW QUESTION 139

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 143

Given: Book.java:

```
public class Book {
private String read(String bname) { return "Read" + bname }
}
```

EBook.java:

```
public class EBook extends Book {
public String read (String url) { return "View" + url }
}
```

Test.java:

```
public class Test {
public static void main (String[] args) { Book b1 = new Book();
b1.read("Java Programing"); Book b2 = new EBook();
b2.read("http://ebook.com/ebook");
}
}
```

What is the result?

- A. Read Java Programming View http:// ebook.com/ebook
- B. Read Java Programming Read http:// ebook.com/ebook
- C. The EBook.java file fails to compile.
- D. The Test.java file fails to compile.

**Answer:** D

#### NEW QUESTION 146

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).reserved().thenComparing(Emp::getlName))`
- B. `.sorted (Comparator.comparing(Emp::getfName).thenComparing(Emp::getlName))`
- C. `.map(Emp::getfName).sorted(Comparator.reserveOrder())`
- D. `.map(Emp::getfName).sorted(Comparator.reserveOrder()).map (Emp::getlName).reserved`

**Answer:** A

#### NEW QUESTION 149

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 150

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 151

Given the Greetings.properties file, containing:

```
HELLO_MSG = Hello, everyone!  
GOODBYE_MSG = Goodbye everyone!
```

and given:

```
import java.util.Enumeration;
import java.util.Locale;
import java.util.ResourceBundle;

public class ResourcesApp {
    public void loadResourceBundle() {
        ResourceBundle resource = ResourceBundle.getBundle("Greetings", Locale.US);
        System.out.println(resource.getObject(1));
    }
    public static void main(String[] args) {
        new ResourcesApp().loadResourceBundle();
    }
}
```

What is the result?

- A. Compilation fails.
- B. GOODBY\_MSG
- C. Hello, everyone!
- D. Goodbye everyone!
- E. HELLO\_MSG

**Answer:** A

#### NEW QUESTION 152

.....

## 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](#)**