

## 1Z0-809 Dumps

### Java SE 8 Programmer II

<https://www.certleader.com/1Z0-809-dumps.html>



**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:

```
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 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:

```
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**

Given the code fragment:

```
Stream<Path> files = Files.walk(Paths.get(System.getProperty("user.home"))); files.forEach (fName -> { //line n1
try {
Path aPath = fName.toAbsolutePath(); //line n2 System.out.println(fName + ":"
+ Files.readAttributes(aPath, Basic.File.Attributes.class).creationTime ());
} catch (IOException ex) { ex.printStackTrace();
});
What is the result?
```

- A. All files and directories under the home directory are listed along with their attributes.
- B. A compilation error occurs at line n1.
- C. The files in the home directory are listed along with their attributes.
- D. A compilation error occurs at line n2.

**Answer:** A

**NEW QUESTION 6**

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 7**

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 8**

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 9

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 10

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 10

Given the code fragment:

```
Path p1 = Paths.get("/Pics/MyPic.jpeg"); System.out.println (p1.getNameCount() + ":" + p1.getName(1) +  
":" + p1.getFileName());
```



Assume that the Pics directory does NOT exist.  
What is the result?

- A. An exception is thrown at run time.
- B. 2:MyPic.jpeg: MyPic.jpeg
- C. 1:Pics:/Pics/ MyPic.jpeg
- D. 2:Pics: MyPic.jpeg

**Answer: B**

#### NEW QUESTION 12

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 17

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 22

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 23

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 26

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 27

Given the code fragment:

```
Connection con = null;  
try {  
    // line n1  
    if(con != null){  
        System.out.print("Connection Established.");  
    }  
  
} catch (Exception e) {  
    System.out.print(e);  
}
```

Assume that dbURL, userName, and password are valid.

Which code fragment can be inserted at line n1 to enable the code to print Connection Established?

- A. Properties prop = new Properties(); prop.put ("user", userName); prop.put ("password", password); con = DriverManager.getConnection (dbURL, prop);
- B. con = DriverManager.getConnection (userName, password, dbURL);
- C. Properties prop = new Properties(); prop.put ("userid", userName); prop.put ("password", password); prop.put("url", dbURL); con = DriverManager.getConnection (prop);
- D. con = DriverManager.getConnection (dbURL); con.setClientInfo ("user", userName); con.setClientInfo ("password", password);

**Answer:** A

### NEW QUESTION 32

Given the structure of the Student table: Student (id INTEGER, name VARCHAR) Given the records from the STUDENT table:

ID	NAME
102	Edwin
103	Edward
103	Edwin

Given the code fragment:

```
Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
Statement st = conn.createStatement();
String query = "DELETE FROM Student WHERE id = 103";
System.out.println("Status: " + st.execute(query));
```

Assume that:

The required database driver is configured in the classpath.

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

- A. The program prints Status: true and two records are deleted from the Student table.
- B. The program prints Status: false and two records are deleted from the Student table.
- C. A SQLException is thrown at runtime.
- D. The program prints Status: false but the records from the Student table are not deleted.

**Answer: B**

#### NEW QUESTION 34

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 35

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 39

Given the code fragments:

```
interface CourseFilter extends Predicate<String> { public default boolean test (String str) {
return str.equals ("Java");
```



```
}  
}  
and  
List<String> strs = Arrays.asList("Java", "Java EE", "Java ME"); Predicate<String> cf1 = s -> s.length() > 3;  
Predicate cf2 = new CourseFilter() { //line n1 public boolean test (String s) {  
return s.contains ("Java");  
}  
};  
long c = strs.stream()  
.filter(cf1)  
.filter(cf2 //line n2  
.count(); System.out.println(c); What is the result?
```

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

**Answer: B**

#### NEW QUESTION 43

Given the structure of the STUDENT table: Student (id INTEGER, name VARCHAR) Given:

```
public class Test {  
static Connection newConnection =null;  
public static Connection get DBConnection () throws SQLException { try (Connection con = DriverManager.getConnection(URL, username, password)) {  
newConnection = con;  
}  
return newConnection;  
}  
public static void main (String [] args) throws SQLException { get DBConnection ();  
Statement st = newConnection.createStatement(); st.executeUpdate("INSERT INTO student VALUES (102, 'Kelvin')");  
}  
}
```

Assume that:

The required database driver is configured in the classpath.

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

What is the result?

- A. The program executes successfully and the STUDENT table is updated with one record.
- B. The program executes successfully and the STUDENT table is NOT updated with any record.
- C. A SQLException is thrown as runtime.
- D. A NullPointerException is thrown as runtime.

**Answer: C**

#### NEW QUESTION 47

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 50

Given:

```
public class Vehicle {  
    int vId;  
    String vName;  
    public Vehicle(int vIdArg, String vNameArg) {  
        this.vId = vIdArg;  
        this.vName = vNameArg;  
    }  
    public int getVId() { return vId; }  
    public String getVName() { return vName; }  
    public String toString() {  
        return vName;  
    }  
}
```

and the code fragment:

```
List<Vehicle> vehicle = Arrays.asList(  
    new Vehicle(2, "Car"),  
    new Vehicle(3, "Bike"),  
    new Vehicle(1, "Truck"));  
vehicle.stream()  
    // line n1  
    .forEach(System.out::print);
```

Which two code fragments, when inserted at line n1 independently, enable the code to print TruckCarBike?

- A. .sorted ((v1, v2) -> v1.getVId() < v2.getVId())
- B. .sorted (Comparable.comparing (Vehicle: :getVName)).reversed ()
- C. .map (v -> v.getVId()).sorted ()
- D. .sorted((v1, v2) -> Integer.compare(v1.getVId(), v2.getVId()))
- E. .sorted(Comparator.comparing ((Vehicle v) -> v.getVId()))

**Answer:** B

#### NEW QUESTION 55

Given:

```
class CheckClass {  
    public static int checkValue (String s1, String s2) { return s1.length() – s2.length();  
    }  
}
```

and the code fragment:

```
String[] strArray = new String [] {"Tiger", "Rat", "Cat", "Lion"}  
//line n1  
for (String s : strArray) { System.out.print (s + " ");  
}
```

Which code fragment should be inserted at line n1 to enable the code to print Rat Cat Lion Tiger?

- A. Arrays.sort(strArray, CheckClass : : checkValue);
- B. Arrays.sort(strArray, (CheckClass : : new) : : checkValue);
- C. Arrays.sort(strArray, (CheckClass : : new).checkValue);
- D. Arrays.sort(strArray, CheckClass : : new : : checkValue);

**Answer:** A

#### NEW QUESTION 56

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 61

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 64

Given the code fragment:

```
LocalDate valentinesDay =LocalDate.of(2015, Month.FEBRUARY, 14);
LocalDate nextYear = valentinesDay.plusYears(1);
nextYear.plusDays(15); //line n1
System.out.println(nextYear);
```

What is the result?

- A. 2016-02-14
- B. A DateTimeException is throw
- C. 2016-02-29
- D. A compilation error occurs at line n1.

**Answer:** A

#### NEW QUESTION 66

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 71

Given the definition of the Country class: `public class country {  
public enum Continent {ASIA, EUROPE} String name;  
Continent region;  
public Country (String na, Continent reg) { name = na, region = reg;  
}  
public String getName () {return name;} public Continent getRegion () {return region;}  
}`  
and the code fragment:  
`List<Country> couList = Arrays.asList (  
new Country ("Japan", Country.Continent.ASIA), new Country ("Italy", Country.Continent.EUROPE),  
new Country ("Germany", Country.Continent.EUROPE)); Map<Country.Continent, List<String>> regionNames = couList.stream ()  
.collect(Collectors.groupingBy (Country ::getRegion, Collectors.mapping(Country::getName, Collectors.toList()))); System.out.println(regionNames);`

- A. {EUROPE = [Italy, Germany], ASIA = [Japan]}
- B. {ASIA = [Japan], EUROPE = [Italy, Germany]}
- C. {EUROPE = [Germany, Italy], ASIA = [Japan]}
- D. {EUROPE = [Germany], EUROPE = [Italy], ASIA = [Japan]}

**Answer:** B

#### NEW QUESTION 74

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 75

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 80

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()  
.forEach(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 81

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 86

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 90

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 92

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 94

Given:

```
final class Folder { //line n1
//line n2
    public void open () { System.out.print("Open");
    }
}
public class Test {
    public static void main (String [] args) throws Exception { try (Folder f = new Folder()) {
```

- A. `f.open();}}`Which two modifications enable the code to print Open Close? (Choose two.)
- B. Replace line n1 with:`class Folder implements AutoCloseable {`
- C. Replace line n1 with:`class Folder extends Closeable {`
- D. Replace line n1 with:`class Folder extends Exception {`
- E. At line n2, insert: `final void close () {System.out.print("Close");}`
- F. At line n2, insert:`public void close () throws IOException { System.out.print("Close");}`

**Answer:** AE

#### NEW QUESTION 95

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 100**

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 104**

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 108**

Given the content:

MessagesBundle.properties file:

```
username = Enter User Name
password = Enter Password
```

MessagesBundle\_fr\_FR.properties file:

```
username = Entrez le nom d'utilisateur
password = Entrez le mot de passe
```

and the code fragment:

```
Locale currentLocale = new Locale.Builder().setRegion("FR").setLanguage("fr").build();
ResourceBundle messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);
Enumeration<String> names = messages.getKeys();
while (names.hasMoreElements()) {
    String key = names.nextElement();
    String name = messages.getString(key);
    System.out.println(key + " = " + name);
}
```

What is the result?

- A. username = Entrez le nom d'utilisateur password = Entrez le mot de passe
- B. username = Enter User Name password = Enter Password
- C. A compilation error occurs.
- D. The program prints nothing.

**Answer: A**

**NEW QUESTION 113**

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 117

Given the content of Operator.java, EngineOperator.java, and Engine.java files:

Operator.java:

```
public abstract class Operator {
    protected void turnON();
    protected void turnOFF();
}
```

EngineOperator.java:

```
public class EngineOperator extends Operator{
    public final void turnON() { System.out.print("ON "); }
    public final void turnOFF() { System.out.println("OFF"); }
}
```

Engine.java:

```
public class Engine{
    Operator m = new EngineOperator();
    public void operate() {
        m.turnON();
        m.turnOFF();
    }
}
```

and the code fragment:

```
Engine carEngine = new Engine();
carEngine.operate();
```

What is the result?

- A. The Engine.java file fails to compile.
- B. The EngineOperator.java file fails to compile.
- C. The Operator.java file fails to compile.
- D. ON OFF

**Answer:** A

#### NEW QUESTION 122

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 124

Given the code fragment:

```
LocalTime now = LocalTime.now();
long timeToBreakfast = 0;
LocalTime office_start = LocalTime.of(7, 30);
if (office_start.isAfter(now)) {
    timeToBreakfast = now.until(office_start, MINUTES);
} else {
    timeToBreakfast = now.until(office_start, HOURS);
}
System.out.println(timeToBreakfast);
```

Assume that the value of now is 6:30 in the morning. What is the result?

- A. An exception is thrown at run time.
- B. 60
- C. 1

**Answer:** D

#### NEW QUESTION 129

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 (newCallerThread("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 130

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 134

Given the code fragments:

```
class Person // line n1
{
    String name;
    Person(String name) {
        this.name = name;
    }
    // line n2
}
```

and

```
List<Person> emps = new ArrayList<>();  
/* code that adds objects of the Person class to the emps list goes here */  
Collections.sort(emps);
```

Which two modifications enable to sort the elements of the emps list? (Choose two.)

- A. Replace line n1 with `class Person extends Comparator<Person>`
- B. At line n2 insert `public int compareTo (Person p) { return this.name.compareTo (p.name);}`
- C. Replace line n1 with `class Person implements Comparable<Person>`
- D. At line n2 insert `public int compare (Person p1, Person p2) { return p1.name.compareTo (p2.name);}`
- E. At line n2 insert `public int compareTo (Person p, Person p2) { return p1.name.compareTo (p2.name);}`
- F. Replace line n1 with `class Person implements Comparator<Person>`

**Answer:** CE

#### NEW QUESTION 135

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 139

Given the code fragment:

```
public static void main (String [ ] args) throws IOException {  
BufferedReader br = new BufferedReader (new InputStreamReader (System.in)); System.out.print ("Enter GDP: ");  
//line 1  
}
```

Which code fragment, when inserted at line 1, enables the code to read the GDP from the user?

- A. `int GDP = Integer.parseInt (br.readLine());`
- B. `int GDP = br.read();`
- C. `int GDP = br.nextInt();`
- D. `int GDP = Integer.parseInt (br.next());`

**Answer:** A

#### NEW QUESTION 142

Given:

```
class Worker extends Thread { CyclicBarrier cb;  
public Worker(CyclicBarrier cb) { this.cb = cb; } public void run () {  
try { cb.await();  
System.out.println("Worker...");  
} catch (Exception ex) { }  
}  
}  
class Master implements Runnable { //line n1 public void run () { System.out.println("Master...");  
}  
}
```

and the code fragment:

```
Master master = new Master();  
//line n2
```

```
Worker worker = new Worker(cb); worker.start();
```

You have been asked to ensure that the run methods of both the Worker and Master classes are executed. Which modification meets the requirement?

- A. At line n2, insert `CyclicBarrier cb = new CyclicBarrier(2, master);`
- B. Replace line n1 with `class Master extends Thread {`
- C. At line n2, insert `CyclicBarrier cb = new CyclicBarrier(1, master);`
- D. At line n2, insert `CyclicBarrier cb = new CyclicBarrier(master);`

**Answer:** C

#### NEW QUESTION 147

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 152

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 153

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 154

Given the code fragment:

```
List<Integer> li = Arrays.asList(10, 20, 30);  
Function<Integer, Integer> fn = f1 -> f1 + f1;  
Consumer<Integer> conVal = s -> System.out.print("Val:" + s + " ");  
li.stream().map(fn).forEach(conVal);
```

What is the result?

- A. Val:20 Val:40 Val:60

- B. Val:10 Val:20 Val:30  
C. A compilation error occurs.  
D. Val: Val: Val:

**Answer:** B

#### NEW QUESTION 158

Given:

```
class Person {  
    String name;  
    int age;  
    public Person(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
    public String getName(){ return name; }  
    public int getAge(){ return age; }  
}
```

and the code fragment:

```
List<Person> sts = Arrays.asList(  
    new Person("Jack", 30),  
    new Person("Mike Hill", 21),  
    new Person("Thomas Hill", 24));  
Stream<Person> resList = sts.stream().filter(s -> s.getAge() >= 25);    // line n1  
long count = resList.filter(s -> s.getName().contains("Hill")).count();  
System.out.print(count);
```

What is the result?

- A. A compilation error occurs at line n1.  
B. An Exception is thrown at run time.  
C. 2

**Answer:** B

#### NEW QUESTION 159

.....



## Thank You for Trying Our Product

\* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

\* One year free update

You can enjoy free update one year. 24x7 online support.

\* Trusted by Millions

We currently serve more than 30,000,000 customers.

\* Shop Securely

All transactions are protected by VeriSign!

**100% Pass Your 1Z0-809 Exam with Our Prep Materials Via below:**

<https://www.certleader.com/1Z0-809-dumps.html>