

# Oracle

## Exam Questions 1Z0-819

Java SE 11 Developer



### NEW QUESTION 1

Given:

```
public class Tester {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder(5);
        sb.append("HOWDY");
        sb.insert(0, ' ');
        sb.replace(3, 5, "LL");
        sb.insert(6, "COW");
        sb.delete(2, 7);
        System.out.println(sb.length());
    }
}
```

What is the result?

- A. 4
- B. 3
- C. An exception is thrown at runtime.
- D. 5

**Answer: D**

**Explanation:**

```
6 public class Tester {
7     public static void main(String[] args) {
8         StringBuilder sb = new StringBuilder (5);
9         sb.append ("HOWDY") ;
10        sb.insert (0, ' ');
11        sb.replace(3, 5, "LL");
12        sb.insert (6, "COW");
13        sb.delete(2, 7);
14        System.out.println(sb.length());
15    }
16 }
```

(command line arguments)

COMPILE & EXECUTE

PASTE SOURCE

Successfully compiled /tmp/java\_82Tlan/Tester.java <-- main method

5

### NEW QUESTION 2

Given:

```
package b;
public class Person {
    protected Person() {                //line 1
    }
}
```

and

```
package a;
import b.Person;
public class Main {                    //line 2
    public static void main(String[] args) {
        Person person = new Person(); //line 3
    }
}
```

Which two allow a.Main to allocate a new Person? (Choose two.)

- A. In Line 1, change the access modifier to privateprivate Person() {
- B. In Line 1, change the access modifier to publicpublic Person() {
- C. In Line 2, add extends Person to the Main classpublic class Main extends Person {and change Line 3 to create a new Main objectPerson person = new Main();
- D. In Line 2, change the access modifier to protectedprotected class Main {
- E. In Line 1, remove the access modifierPerson() {

**Answer: BC**

### NEW QUESTION 3

Given:

```
public class Tester {  
    private int x;  
    private static int y;  
    public static void main(String[] args) {  
        Tester t1 = new Tester();  
        t1.x = 2;  
        Tester.y = 3;  
        Tester t2 = new Tester();  
        t2.x = 4;  
        t2.y = 5;  
        System.out.println(t1.x+", "+t1.y);  
        System.out.println(t2.x+", "+Tester.y);  
        System.out.println(t2.x+", "+t1.y);  
    }  
}
```

What is the result?

- A. 2,34,34,5
- B. 2,34,54,5
- C. 2,54,54,5
- D. 2,34,54,3

Answer: C

Explanation:



#### NEW QUESTION 4

Given the code fragment:

```
int x = 0;  
while(x < 10){  
    System.out.print(x++);  
}
```

Which "for" loop produces the same output?

- A.
- ```
int b = 0;
for( ; b < 10; ){
    System.out.print(++b);
}
```
- B.
- ```
for(a; a < 10; a++){
    System.out.print(a);
}
```
- C.
- ```
for(int d = 0; d < 10; ){
    System.out.print(d);
    ++d;
}
```
- D.
- ```
for(int c = 0; ; c++){
    System.out.print(c);
    if(c == 10){
        break;
    }
}
```

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

**Answer: C**

#### NEW QUESTION 5

Given:

```
public static void main(String[] args) {
    final List<String> fruits =
        List.of("Orange", "Apple", "Lemmon", "Raspberry");
    final List<String> types =
        List.of("Juice", "Pie", "Ice", "Tart");
    final var stream =
        IntStream.range(0, Math.min(fruits.size(), types.size()))
            .mapToObj((i) -> fruits.get(i) + " " + types.get(i) );
    stream. forEach(System.out::println);
}
```

What is the result?

- A. Orange Juice  
B. The compilation fails.  
C. Orange Juice Apple Pie Lemmon Ice Raspberry Tart  
D. The program prints nothing.

**Answer: C**

**Explanation:**

```

12 public class Person {
13     public static void main (String[] args) {
14         final List<String> fruits =
15             List.of("Orange", "Apple", "Lemmon", "raspberry");
16         final List<String> types =
17             List.of("Juice", "Pie", "Ice", "Tart");
18         final var stream =
19             IntStream.range(0, Math.min(fruits.size(), types.size()))
20                 .mapToObj ((i) -> fruits.get(i) + " " + types.get(i) );
21         stream. forEach(System.out::println);
22     }
23 }
24 }

```

#### Result

compiled and executed in 1.227 sec(s)

```

Orange Juice
Apple Pie
Lemmon Ice
raspberry Tart

```

#### NEW QUESTION 6

Given:

```

public class Foo {
    public <T> Collection<T> foo(Collection<T> arg) { ... }
}

```

and

```

public class Bar extends Foo { ... }

```

Which two statements are true if the method is added to Bar? (Choose two.)

- A. public Collection<String> foo(Collection<String> arg) { ... } overrides Foo.foo.
- B. public <T> Collection<T> foo(Stream<T> arg) { ... } overloads Foo.foo.
- C. public <T> List<T> foo(Collection<T> arg) { ... } overrides Foo.foo.
- D. public <T> Collection<T> foo(Collection<T> arg) { ... } overloads Foo.foo.
- E. public <T> Collection<T> bar(Collection<T> arg) { ... } overloads Foo.foo.
- F. public <T> Iterable<T> foo(Collection<T> arg) { ... } overrides Foo.foo.

**Answer:** CF

#### NEW QUESTION 7

Given:

```
public class A {  
    private boolean checkValue(int val) {  
        return true;  
    }  
}
```

and

```
public class B extends A {  
    public int modifyVal(int val) {  
        if(checkValue(val)) {  
            return val;  
        } else {  
            return 0;  
        }  
    }  
    public static void Main(String[] args) {  
        B b = new B();  
        System.out.println(b.modifyVal(10));  
    }  
}
```

What is the result?

- A. nothing
- B. It fails to compile.
- C. A java.lang.IllegalArgumentException is thrown.
- D. 10

**Answer:** B

**Explanation:**

```

1- public class A {
2-     private boolean checkValue(int val) {
3-         return true;
4-     }
5- }
6- and
7- public class B extends A {
8-     public int modifyVal(int val) {
9-         if(checkValue(val)) {
10-             return val;
11-         } else {
12-             return 0;
13-         }
14-     }
15-     public static void Main(String[] args) {
16-         B b = new B();
17-         system.out.println(b.modfiyVal (10));
18-     }
19- }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: sec(s), Memory: kilobyte(s)

```

/A.java:6: error: class, interface, or enum expected
and
^
1 error

```

#### NEW QUESTION 8

Given:

```

1. {
2.     Iterator iter = List.of(1,2,3).iterator();
3.     while (iter.hasNext()) {
4.         foo(iter.next());
5.     }
6.     Iterator iter2 = List.of(1,2,3).iterator();
7.     while (iter.hasNext()) {
8.         bar(iter2.next());
9.     }
10. }
11. for (Iterator iter = List.of(1,2,3).iterator(); iter.hasNext(); ) {
12.     foo(iter.next());
13. }
14. for (Iterator iter2 = List.of(1,2,3).iterator(); iter.hasNext(); ) {
15.     bar(iter2.next());
16. }

```

Which loop incurs a compile time error?

- A. the loop starting line 11
- B. the loop starting line 7
- C. the loop starting line 14
- D. the loop starting line 3

Answer: C

#### NEW QUESTION 9

Examine this excerpt from the declaration of the java.se module:

```
module java.se {  
    ...  
    requires transitive java.sql;  
    ...  
}
```

What does the transitive modifier mean?

- A. Only a module that requires the java.se module is permitted to require the java.sql module.
- B. Any module that requires the java.se module does not need to require the java.sql module.
- C. Any module that attempts to require the java.se module actually requires the java.sql module instead.
- D. Any module that requires the java.sql module does not need to require the java.se module.

**Answer: A**

#### NEW QUESTION 10

Given:

```
public class MethodTest {  
    // line 1  
}
```

Which two method implementations are correct, when inserted independently in line 1? (Choose two.)

A.

```
public boolean methodD(int x) {  
    return x > 0;  
}
```

B.

```
public String methodB() {  
    System.out.println("methodB");  
}
```

C.

```
public char methodE (String msg) {  
    return msg;  
}
```

D.

```
public void methodC(int x) {  
    return ++x;  
}
```

E.

```
public void methodA() {  
    System.out.println("methodA");  
}
```

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

**Answer: AE**

#### NEW QUESTION 10

Given:

```
public class Foo {  
    public void foo(Collection arg) {  
        System.out.println("Bonjour le monde!");  
    }  
}
```

and

```
public class Bar extends Foo {  
    public void foo(Collection arg) {  
        System.out.println("Hello world!");  
    }  
    public void foo(List arg) {  
        System.out.println("Olá Mundo!");  
    }  
}
```

and

```
Foo f1 = new Foo();  
Foo f2 = new Bar();  
Bar b1 = new Bar();  
Collection<String> c = new ArrayList<>();
```

Which three are true? (Choose three.)

- A. b1.foo(c) prints Bonjour le monde!
- B. f1.foo(c) prints Hello world!
- C. f1.foo(c) prints Olá Mundo!
- D. b1.foo(c) prints Hello world!
- E. f2.foo(c) prints Olá Mundo!
- F. b1.foo(c) prints Olá Mundo!
- G. f2.foo(c) prints Bonjour le monde!
- H. f2.foo(c) prints Hello world!
- I. f1.foo(c) prints Bonjour le monde!

**Answer:** BFG

#### NEW QUESTION 15

Given:

```
1. interface Pastry {  
2.     void getIngredients();  
3. }  
4. abstract class Cookie implements Pastry {}  
5.  
6. class ChocolateCookie implements Cookie {  
7.     public void getIngredients() {}  
8. }  
9. class CoconutChocolateCookie extends ChocolateCookie {  
10.     void getIngredients(int x) {}  
11. }
```

Which is true?

- A. The compilation fails due to an error in line 6.
- B. The compilation succeeds.
- C. The compilation fails due to an error in line 4.
- D. The compilation fails due to an error in line 10.
- E. The compilation fails due to an error in line 7.
- F. The compilation fails due to an error in line 9.
- G. The compilation fails due to an error in line 2.

**Answer:** A

#### NEW QUESTION 18

Given:

jdeps -jdkinternals C:\workspace4\SimpleSecurity\jar\classes.jar

Which describes the expected output?

- A. jdeps lists the module dependencies and the package names of all referenced JDK internal API
- B. If any are found, the suggested replacements are output in the console.
- C. jdeps outputs an error message that the -jdkinternals option requires either the -summary or the -verbose options to output to the console.
- D. The -jdkinternals option analyzes all classes in the .jar and prints all class-level dependencies.
- E. The -jdkinternals option analyzes all classes in the .jar for class-level dependencies on JDK internal API

F. If any are found, the results with suggested replacements are output in the console.

**Answer:** A

**Explanation:**

-jdkinternals option analyzes all classes in the .jar for class-level dependencies on JDK internal APIs. If any are found, the results with suggested replacements are output in the console.

**NEW QUESTION 20**

Given:

```
int arr[][] = {{5,10},{8,12},{9,3}};
long count = Stream.of(arr)
    .flatMapToInt(IntStream::of)
    .map(n -> n + 1)
    .filter(n -> (n % 2 == 0))
    .peek(System.out::print)
    .count();
System.out.println(" " + count);
```

What is the result?

- A. 6910 3
- B. 10126 3
- C. 3
- D. 6104 3

**Answer:** D

**Explanation:**

```

1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7  import java.util.function.Consumer;
8  import java.util.stream.Stream;
9  import java.util.stream.IntStream;
10
11
12  public class Main {
13
14  public static void main(String[] args) {
15      int arr[][] = {{5,10}, {8,12}, {9,3}};
16      long count = Stream.of(arr)
17          .flatMapToInt(IntStream::of)
18          .map (n -> n + 1)
19          .filter(n -> (n % 2 == 0))
20          .peek(System.out::print)
21          .count();
22      System.out.println(" " + count);
23  }
24  }

```

#### Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

#### CommandLine Arguments

#### Result

CPU Time: 0.32 sec(s), Memory: 34220 kilobyte(s)

6104 3

#### NEW QUESTION 22

Given:

```

public class Tester {
    static class Person implements /* line 1 */ {
        private String name;
        Person(String name) { this.name = name; }
        /* line 2 */
    }
    public static void main(String[] args) {
        Person[] people = {new Person("Joe"),
                           new Person("Jane"),
                           new Person("John")};
        Arrays.sort(people);
        for(Person person: people) {
            System.out.println(person.name);
        }
    }
}

```

You want the code to produce this output:

John

Joe Jane

Which code fragment should be inserted on line 1 and line 2 to produce the output?

- A. Insert `Comparator<Person>` on line 1. Insert `public int compare(Person p1, Person p2) { return p1.name.compare(p2.name);}` on line 2.
- B. Insert `Comparator<Person>` on line 1. Insert `public int compareTo(Person person) { return person.name.compareTo(this.name);}` on line 2.

- C. Insert Comparable<Person> on line 1.Insertpublic int compare(Person p1, Person p2) { return p1.name.compare(p2.name);}on line 2.  
D. Insert Comparator<Person> on line 1.Insertpublic int compare(Person person) { return person.name.compare(this.name);}on line 2.

**Answer: B**

#### NEW QUESTION 25

Given:

```
public class Main {  
    class Student {                                // line 1  
        String classname;  
        Student(String classname) {                // line 2  
            this.classname = classname;  
        }  
    }  
    public static void main(String[] args) {  
        var student = new Student("Biology"); // line 3  
    }  
}
```

Which two independent changes will make the Main class compile? (Choose two.)

- A. Move the entire Student class declaration to a separate Java file, Student.java.  
B. Change line 2 to public Student(String classname).  
C. Change line 1 to public class Student {.  
D. Change line 3 to Student student = new Student("Biology");.  
E. Change line 1 to static class Student {.

**Answer: BD**

**Explanation:**

```
1  import java.util.*;  
2  import java.io.*;  
3  import java.lang.Thread;  
4  import java.util.ArrayList;  
5  import java.util.LinkedList;  
6  import java.util.List;  
7  import java.util.function.Consumer;  
8  import java.util.stream.Stream;  
9  import java.util.stream.IntStream;  
10 import java.util.Optional;  
11  
12  
13 - public class Main {  
14 -     class Student {  
15         String classname;  
16 -         public Student (String classname) {  
17             this.classname = classname;  
18         }  
19  
20     }  
21 -     public static void main (String[] args) {  
22         var student = new Student ("Biology");  
23     }  
24 }
```

#### NEW QUESTION 27

Given:

```
public class SerializedMessage implements Serializable {
    String message;
    LocalDateTime createdTime;
    transient LocalDateTime updatedDateTime;;
    SerializedMessage(String message) {
        this.message = message;
        this.createdTime = LocalDateTime.now();
    }
    private void readObject (ObjectInputStream in) {
        try {
            in.defaultReadObject();
            this.updatedDateTime = LocalDateTime.now();
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

When is the readObject method called?

- A. before this object is deserialized
- B. after this object is deserialized
- C. before this object is serialized
- D. The method is never called.
- E. after this object is serialized

**Answer:** B

#### NEW QUESTION 32

Given the code fragment:

```
var pool = Executors.newFixedThreadPool(5);
```

```
Future outcome = pool.submit(() > 1);
```

Which type of lambda expression is passed into submit()?

- A. java.lang Runnable
- B. java.util.function.Predicate
- C. java.util.function.Function
- D. java.util.concurrent.Callable

**Answer:** D

#### NEW QUESTION 37

Given:

```
public interface A {
    abstract void x();
}
```

and

```
public abstract class B /* position 1 */ {
    /* position 2 */
    public void x() { }
    public abstract void z();
}
```

and

```
public class C extends B implements A {
    /* position 3 */
}
```

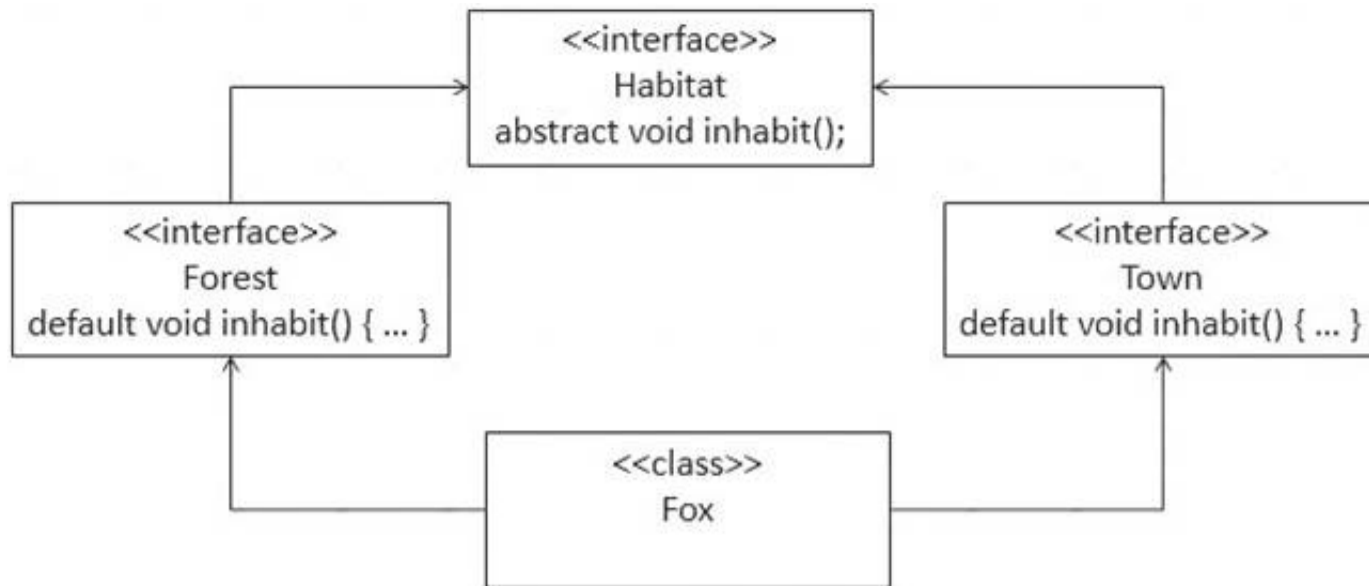
Which code, when inserted at one or more marked positions, would allow classes B and C to compile?

- A. @Override // position 3 void x () {} // position 3 @Override // position 3 public void z() {} // position 3
- B. @Override // position 2 public void z() {} // position 3
- C. implements A // position 1 @Override // position 2
- D. public void z() {} // position 3

**Answer:** B

#### NEW QUESTION 40

Given:



Which statement is true about the Fox class?

- A. Fox class does not have to override inhabit method, so long as it does not try to call it.
- B. Fox class does not have to override the inhabit method if Forest and Town provide compatible implementations.
- C. Fox class must implement either Forest or Town interfaces, but not both.
- D. The inhabit method implementation from the first interface that Fox implements will take precedence.
- E. Fox class must provide implementation for the inhabit method.

**Answer: B**

#### NEW QUESTION 41

Given:

```

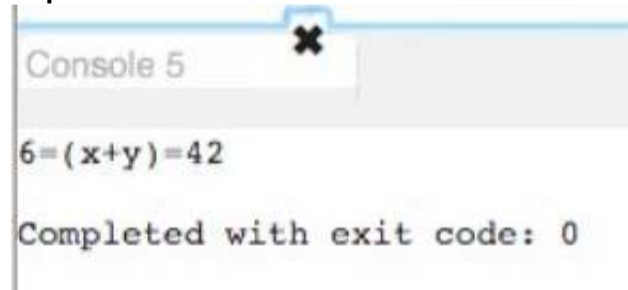
public class Tester {
    public static void main(String[] args) {
        int x = 4;
        int y = 2;
        System.out.println(x+y+"=(x+y)="+x+y);
    }
}
    
```

What is the result?

- A. An exception is thrown at runtime
- B. 42=(x+y)=42
- C. 42=(x+y)=6
- D. 6=(x+y)=42
- E. 6=(x+y)=6

**Answer: D**

**Explanation:**



#### NEW QUESTION 45

Given the code fragment:

```

public static void main(String[] args) {
    List<Integer> even = List.of();
    even.add(0, -1);
    even.add(0, -2);
    even.add(0, -3);
    System.out.println(even);
}
    
```

What is the output?

- A. The compilation fail
- B. [-1, -2, -3]
- C. [-3, -2, -1]
- D. A runtime exception is thrown.

**Answer: D**

**NEW QUESTION 48**

Which describes a characteristic of setting up the Java development environment?

- A. Setting up the Java development environment requires that you also install the JRE.
- B. The Java development environment is set up for all operating systems by default.
- C. You set up the Java development environment for a specific operating system when you install the JDK.
- D. Setting up the Java development environment occurs when you install an IDE before the JDK.

**Answer:** D

**NEW QUESTION 53**

Given:

```
class Myclass {
public static void main(String [] args) {
    System.out.println(arg[1] + "--" + arg[3] + "--" + arg[0]);
}
}
```

executed using this command: java Myclass My Car is red What is the output of this class?

- A. Car--red--My
- B. My--Car--is
- C. My--is--java
- D. java--Myclass--My
- E. Myclass--Car--red

**Answer:** A

**NEW QUESTION 56**

What makes Java dynamic?

- A. At runtime, classes are loaded as needed, and new code modules can be loaded on demand.
- B. The runtime can process machine language sources as well as executables from different language compilers.
- C. The Java compiler uses reflection to test if class methods are supported by resources of a target platform.
- D. The Java compiler preprocesses classes to run on specific target platforms.

**Answer:** A

**NEW QUESTION 57**

Given:

```
LocalDate d1 = LocalDate.of(1997,2,7); DateTimeFormatter dtf = DateTimeFormatter.ofPattern( /*insert code here*/ ); System.out.println(dtf.format (d1));
```

Which pattern formats the date as Friday 7th of February 1997?

- A. "eeee dd+"th of"+ MMM yyyy"
- B. "eeee dd'th of' MMM yyyy"
- C. "eeee d+"th of"+ MMMM yyyy"
- D. "eeee d'th of' MMMM yyyy"

**Answer:** B

**NEW QUESTION 61**

Given:

```
public class Test {
    public static void doThings() throws GeneralException {
        try {
            throw new RuntimeException("Someting happened");
        } catch (Exception e) {
            throw new SpecificException(e.getMessage());
        }
    }
    public static void main(String args[]) {
        try{
            Test.doThings();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
class GeneralException /* line 1 */ {
    public GeneralException(String s) { super(s); }
}
class SpecificException /* line 2 */ {
    public SpecificException(String s) { super(s); }
}
```

Which option should you choose to enable the code to print Something happened?

- A. Add extends GeneralException on line 1.Add extends Exception on line 2.
- B. Add extends SpecificException on line 1.Add extends GeneralException on line 2.
- C. Add extends Exception on line 1.Add extends Exception on line 2.
- D. Add extends Exception on line 1.Add extends GeneralException on line 2.

**Answer: D**

**Explanation:**

```
1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7
8  public class Test {
9
10     public static void doThings() throws GeneralException {
11         try{
12             throw new RuntimeException("Something happened");
13         } catch (Exception e) {
14             throw new SpecificException (e.getMessage());
15         }
16     }
17 }
18
19     public static void main(String args[]) {
20         try{
21             Test.doThings();
22         }catch (Exception e) {
23             System.out.println(e.getMessage());
24         }
25     }
26     class GeneralException extends Exception {
27         public GeneralException(String s) { super(s); }
28     }
29     class SpecificException extends GeneralException {
30         public SpecificException(String s) { super(s);}
31     }
32 }
```

#### NEW QUESTION 66

Given:

```
public static void main(String[] args) {
    try (Reader reader1 = new FileReader("File1.txt");
        Reader reader2 = new FileReader("File2.txt");
        Reader reader3 = new FileReader("File3_txt")) {

    } catch (IOException ex) {
        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
    }
    // Line 1
    System.out.println("Done");
}
```

When run and all three files exist, what is the state of each reader on Line 1?

- A. All three readers are still open.
- B. All three readers have been closed.
- C. The compilation fails.
- D. Only reader1 has been closed.

**Answer: C**

#### NEW QUESTION 70

Given:

```
public class Foo {  
    private final ReentrantLock lock = new ReentrantLock();  
    private State state;  
    public void foo() throws Exception {  
        try {  
            lock.lock();  
            state.mutate();  
        }  
        finally {  
            lock.unlock();  
        }  
    }  
}
```

What is required to make the Foo class thread safe?

- A. No change is required.
- B. Make the declaration of lock static.
- C. Replace the lock constructor call with new ReentrantLock (true).
- D. Move the declaration of lock inside the foo method.

**Answer: C**

#### NEW QUESTION 71

Given:

```
import java.io.FileNotFoundException;  
import java.io.IOException;  
  
public class Tester {  
    public static void main(String[] args) {  
        try {  
            doA();  
        } //line 1  
    }  
    private static void doA() throws IOException, IndexOutOfBoundsException {  
        if (false) {  
            throw new FileNotFoundException();  
        } else {  
            throw new IndexOutOfBoundsException();  
        }  
    }  
}
```

What must be added in line 1 to compile this class?

- A. catch(IOException e) {}
- B. catch(FileNotFoundException | IndexOutOfBoundsException e) {}
- C. catch(FileNotFoundException | IOException e) {}
- D. catch(IndexOutOfBoundsException e) {} catch(FileNotFoundException e) {}
- E. catch(FileNotFoundException e) {} catch(IndexOutOfBoundsException e) {}

**Answer: A**

#### NEW QUESTION 75

Given:

```
public class Foo {  
    public void foo(Collection arg) {  
        System.out.println("Bonjour le monde!");  
    }  
}
```

and

```
public class Bar extends Foo {  
    public void foo(Collection arg) {  
        System.out.println("Hello world!");  
    }  
    public void foo(List arg) {  
        System.out.println("Hola Mundo!");  
    }  
}
```

and

```
Foo f1 = new Foo();  
Foo f2 = new Bar();  
Bar b1 = new Bar();  
List<String> li = new ArrayList<>();
```

Which three are correct? (Choose three.)

- A. b1.foo(li) prints Hello world!
- B. f1.foo(li) prints Bonjour le monde!
- C. f1.foo(li) prints Hello world!
- D. f1.foo(li) prints Hola Mundo!
- E. b1.foo(li) prints Bonjour le monde!
- F. f2.foo(li) prints Hola Mundo!
- G. f2.foo(li) prints Bonjour le monde!
- H. b1.foo(li) prints Hola Mundo!
- I. f2.foo(li) prints Hello world!

**Answer:** ABH

#### NEW QUESTION 76

Which is the correct order of possible statements in the structure of a Java class file?

- A. class, package, import
- B. package, import, class
- C. import, package, class
- D. package, class, import
- E. import, class, package

**Answer:** B

#### NEW QUESTION 81

Given:

```
@Target(ElementType.METHOD)  
@Retention(RetentionPolicy.RUNTIME)  
public @interface AuthorInfo {  
    String author() default "";  
    String date();  
    String[] comments() default {};  
}
```

Which two are correct? (Choose two.)

- A. `@AuthorInfo(date="1-1-2020", comments={ null })`  
`public class Hello {`  
 `public void func() {}`  
`}`
- B. `public class Hello {`  
`@AuthorInfo (date="1-1-2020. comments="Hello")`  
 `public void func() {}`  
`}`
- C. `public class Hello {`  
 `@AuthorInfo`  
 `public void func() {}`  
`}`
- D. `@AuthorInfo(date="1-1-2020")`  
`public class Hello {`  
 `public void func() {}`  
`}`
- E. `public class Hello {`  
 `@AuthorInfo(date="1-1-2020", author="Gandhi", comments={ "world" })`  
 `public void func () {}`  
`}`

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

**Answer:** CD

#### NEW QUESTION 86

Given the code fragment:

```
char[][] arrays = {{'a', 'd'}, {'b', 'e'}, {'c', 'f'}};
for (char[] xx : arrays) {
    for (char yy : xx) {
        System.out.print(yy);
    }
    System.out.print(" ");
}
```

What is the result?

- A. ab cd ef  
B. An `ArrayIndexOutOfBoundsException` is thrown at runtime.  
C. The compilation fails.  
D. abc def  
E. ad be cf

**Answer:** E

#### NEW QUESTION 91

You are working on a functional bug in a tool used by your development organization. In your investigation, you find that the tool is executed with a security policy file containing this grant.

```
grant codebase "file:${klib.home}/j2se/home/klib.jar" {
    permission java.security.AllPermission;
};
```

What action should you take?

- A. Nothing, because it is an internal tool and not exposed to the public.  
B. Remove the grant because it is excessive.  
C. Nothing, because it is not related to the bug you are investigating.  
D. File a security bug against the tool referencing the excessive permission granted.  
E. Nothing, because listing just the required permissions would be an ongoing maintenance challenge.

**Answer:** D

#### NEW QUESTION 94

Given the code fragment:

```
int[] secA = { 2, 4, 6, 8, 10 };
int[] secB = { 2, 4, 8, 6, 10 };
int res1 = Arrays.mismatch(secA, secB);
int res2 = Arrays.compare(secA, secB);
System.out.print(res1 + " : " + res2);
```

What is the result?

- A. -1 : 2
- B. 2 : -1
- C. 2 : 3
- D. 3 : 0

**Answer:** B

#### NEW QUESTION 98

Given:

```
String[][] arr = {
    {"Red", "White"},
    {"Black"},
    {"Blue", "Yellow", "Green", "Violet"}
};
for(int row = 0; row < arr.length; row++) {
    int column = 0;
    for(; column < arr[row].length; column++) {
        System.out.println "[" + row + ", " + column + "] = " + arr[row][column]);
    }
}
```

What is the result?

- A. [0,0] = Red[0,1] = White[1,0] = Black[1,1] = Blue[2,0] = Yellow[2,1] = Green[3,0] = Violet
- B. [0,0] = Red[1,0] = Black[2,0] = Blue
- C. java.lang.ArrayIndexOutOfBoundsException thrown
- D. [0,0] = Red[0,1] = White[1,0] = Black[2,0] = Blue[2,1] = Yellow[2,2] = Green[2,3] = Violet

**Answer:** D

**Explanation:**



#### NEW QUESTION 103

Examine these module declarations:

```
module ServiceAPI {
    exports com.example.api;
}

module ServiceProvider {
    requires ServiceAPI;
    provides com.example.api with com.myimpl.Impl;
}

module Consumer {
    requires ServiceAPI;
    uses com.example.api;
}
```

Which two statements are correct? (Choose two.)

- A. The ServiceProvider module is the only module that, at run time, can provide the com.example.api API.
- B. The placement of the com.example.api API in a separate module, ServiceAPI, makes it easy to install multiple provider modules.
- C. The Consumer module should require the ServiceProvider module.
- D. The ServiceProvider module should export the com.myimpl package.
- E. The ServiceProvider module does not know the identity of a module (such as Consumer) that uses the com.example.api API.

**Answer:** AC

#### NEW QUESTION 104

Which two describe reasons to modularize the JDK? (Choose two.)

- A. easier to understand the Java language
- B. improves security and maintainability
- C. easier to expose implementation details
- D. improves application robustness
- E. easier to build a custom runtime linking application modules and JDK modules

**Answer:** BD

#### NEW QUESTION 107

Given the code fragment:

```
int x = 0;
do {
    x++;
    if (x == 1) {
        continue;
    }
    System.out.println(x);
} while(x < 1);
```

What is the result?

- A. 01
- B. 1
- C. The program prints nothing.
- D. It prints 1 in the infinite loop.

**Answer:** D

#### NEW QUESTION 111

Given:

```
public interface TestInterface {
    default void samplingProbeProcedure() {
        probeProcedure();
        System.out.println("Collect Sample");
        System.out.println("Leave Asteroid");
        System.out.println("Dock with Main Craft");
    }
    default void explosionProbeProcedure() {
        probeProcedure();
        System.out.println("Explode")
    }
}
```

Examine these requirements:

- Eliminate code duplication.
- Keep constant the number of methods other classes may implement from this interface. Which method can be added to meet these requirements?

```
A. private default void probeProcedure(){
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}

B. static void probeProcedure(){
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}

C. private void probeProcedure(){
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}

D. default void probeProcedure(){
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}
```

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

**Answer:** B

#### NEW QUESTION 114

Which two statements correctly describe capabilities of interfaces and abstract classes? (Choose two.)

- A. Interfaces cannot have protected methods but abstract classes can.
- B. Both interfaces and abstract classes can have final methods.
- C. Interfaces cannot have instance fields but abstract classes can.
- D. Interfaces cannot have static methods but abstract classes can.
- E. Interfaces cannot have methods with bodies but abstract classes can.

**Answer:** AC

#### NEW QUESTION 119

Given:

```
void myLambda() {
    int i = 25;
    Supplier<Integer> foo = () -> i;
    i++;
    System.out.println(foo.get());
}
```

Which is true?

- A. The code compiles but does not print any result.
- B. The code prints 25.
- C. The code does not compile.
- D. The code throws an exception at runtime.

**Answer:** C

#### NEW QUESTION 122

Given:

```
public class Hello {
    class Greeting {
        void sayHi() {
            System.out.println("Hello world");
        }
    }
    public static void main(String... args) {
        // Line 1
    }
}
```

What code must you insert on Line 1 to enable the code to print Hello world?

- A. Hello.Greeting myG = new Hello.Greeting() myG.sayHi();

- B. Hello myH = new Hello();Hello.Greeting myG = myH.new Greeting(); myG.sayHi();  
 C. Hello myH = new Hello();Hello.Greeting myG = myH.new Hello.Greeting(); myG.sayHi();  
 D. Hello myH = new Hello(); Greeting myG = new Greeting(); myG.sayHi ();

**Answer:** B

#### NEW QUESTION 124

Which two statements are correct about modules in Java? (Choose two.)

- A. java.base exports all of the Java platforms core packages.  
 B. module-info.java can be placed in any folder inside module-path.  
 C. A module must be declared in module-info.java file.  
 D. module-info.java cannot be empty.  
 E. By default, modules can access each other as long as they run in the same folder.

**Answer:** AC

#### NEW QUESTION 129

Given:

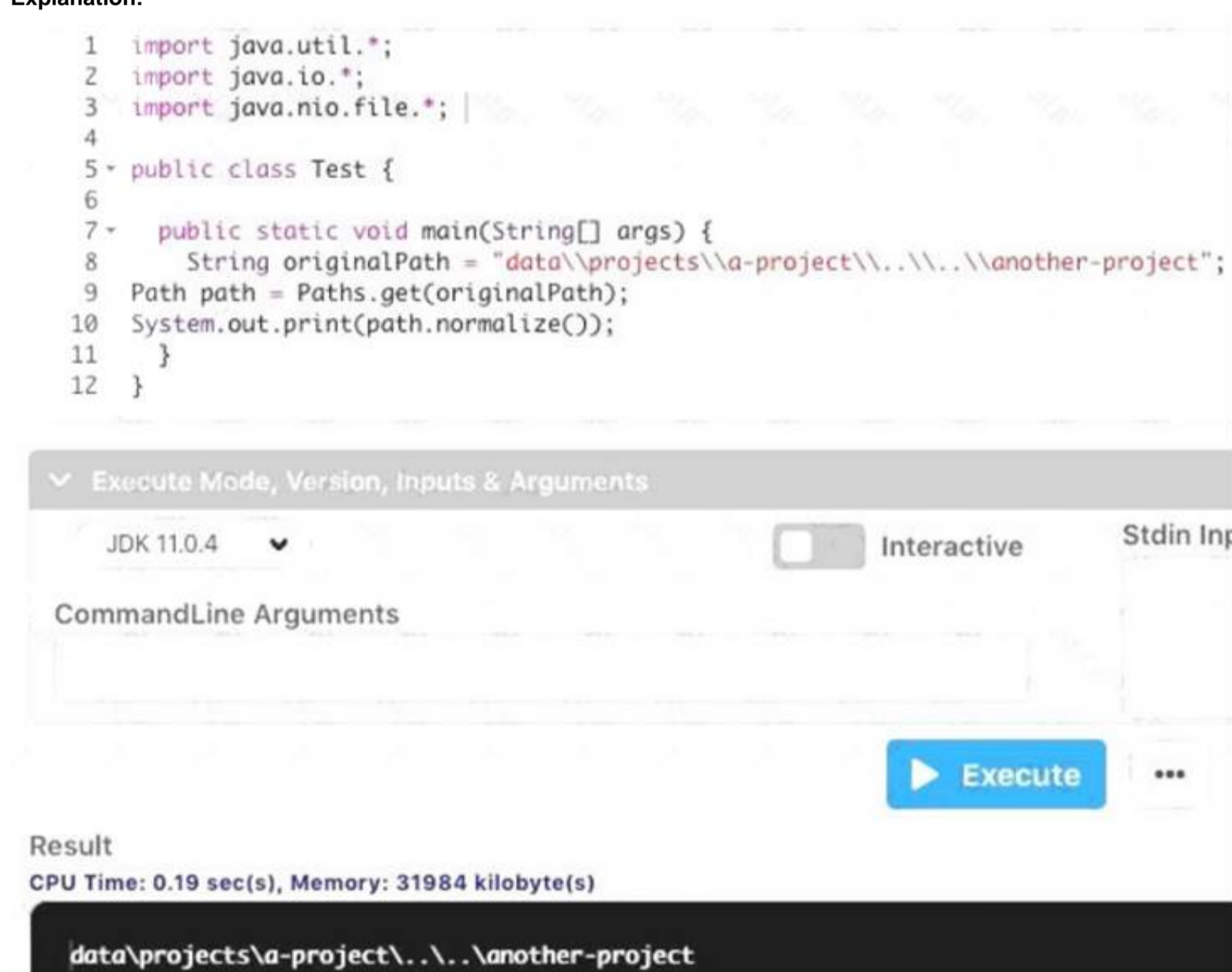
String originalPath = "data\\projects\\a-project\\..\\..\\another-project"; Path path = Paths.get(originalPath); System.out.print(path.normalize());

What is the result?

- A. data\\another-project  
 B. data\\projects\\a-project\\another-project  
 C. data\\projects\\a-project\\..\\..\\another-project  
 D. data\\projects\\a-project\\..\\..\\another-project

**Answer:** D

**Explanation:**



```

1  import java.util.*;
2  import java.io.*;
3  import java.nio.file.*;
4
5  public class Test {
6
7      public static void main(String[] args) {
8          String originalPath = "data\\projects\\a-project\\..\\..\\another-project";
9          Path path = Paths.get(originalPath);
10         System.out.print(path.normalize());
11     }
12 }
    
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4 ☐ Interactive Stdin Input

CommandLine Arguments

**Execute**

**Result**  
 CPU Time: 0.19 sec(s), Memory: 31984 kilobyte(s)

data\\projects\\a-project\\..\\..\\another-project

#### NEW QUESTION 134

Assume ds is a DataSource and the EMP table is defined appropriately.

```
try (Connection conn = ds.getConnection();
    PreparedStatement ps = conn.prepareStatement("INSERT INTO EMP VALUES(?, ?, ?)") ) {
    ps.setObject(1, 101, JDBCType.INTEGER);
    ps.setObject(2, "SMITH", JDBCType.VARCHAR);
    ps.setObject(3, "HR", JDBCType.VARCHAR);
    ps.executeUpdate();
    ps.setInt(1, 102);
    ps.setString(2, "JONES");
    ps.executeUpdate();
}
```

What does executing this code fragment do?

- A. inserts two rows (101, 'SMITH', 'HR') and (102, 'JONES', NULL)
- B. inserts two rows (101, 'SMITH', 'HR') and (102, 'JONES', 'HR')
- C. inserts one row (101, 'SMITH', 'HR')
- D. throws a SQLException

**Answer: C**

#### NEW QUESTION 135

Given:

```
public class Main {
    public static void main(String[] args) {
        int i = 1;
        for(String s : args) {
            System.out.println((i++) + " " + s);
        }
    }
}
```

executed with this command: java Main one two three

What is the output of this class?

- A. The compilation fails.
- B. 1) one2) two3) three
- C. A java.lang.ArrayIndexOutOfBoundsException is thrown.
- D. 1) one
- E. nothing

**Answer: B**

#### NEW QUESTION 140

Given:

```
try {
    // line 1
    lines.map(l -> l.toUpperCase())
        .forEach (line --> {
            try {
                Files.write(Paths.get("outputFile_to_path"),
line.getBytes(), StandardOpenOption.CREATE);
            } catch (IOException e) {
                e.printStackTrace();
            }
        });
} catch (IOException e) {
    e.printStackTrace();
}
```

You want to obtain the Stream object on reading the file. Which code inserted on line 1 will accomplish this?

- A. var lines = Files.lines(Paths.get(INPUT\_FILE\_NAME));
- B. Stream lines = Files.readAllLines(Paths.get(INPUT\_FILE\_NAME));
- C. var lines = Files.readAllLines(Paths.get(INPUT\_FILE\_NAME));
- D. Stream<String> lines = Files.lines(INPUT\_FILE\_NAME);

**Answer: C**

#### NEW QUESTION 141

Given:

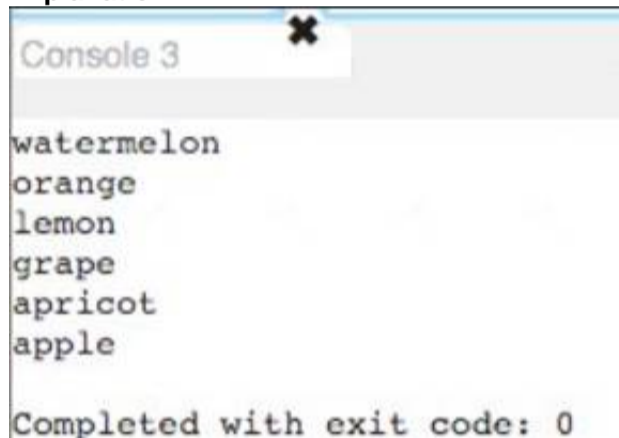
```
import java.util.ArrayList;
import java.util.Arrays;
public class NewMain {
    public static void main(String[] args) {
        String[] fruitNames = { "apple", "orange",
                                "grape", "lemon", "apricot", "watermelon" };
        var fruits = new ArrayList<>(Arrays.asList(fruitNames));
        fruits.sort((var a, var b) -> -a.compareTo(b));
        fruits.forEach(System.out::println);
    }
}
```

What is the result?

- A. watermelonorangelemongrapeapricotapple
- B. nothing
- C. appleapricotgrapelemonorangewatermelon
- D. appleorangegrapelemonapricotwatermelon

**Answer:** A

**Explanation:**



```
Console 3
watermelon
orange
lemon
grape
apricot
apple
Completed with exit code: 0
```

#### NEW QUESTION 146

Which two are successful examples of autoboxing? (Choose two.)

- A. String a = "A";
- B. Integer e = 5;
- C. Float g = Float.valueOf(null);
- D. Double d = 4;
- E. Long c = 23L;
- F. Float f = 6.0;

**Answer:** AB

#### NEW QUESTION 149

Which describes an aspect of Java that contributes to high performance?

- A. Java prioritizes garbage collection.
- B. Java has a library of built-in functions that can be used to enable pipeline burst execution.
- C. Java monitors and optimizes code that is frequently executed.
- D. Java automatically parallelizes code execution.

**Answer:** C

#### NEW QUESTION 152

Given:

```
public class Main {
    public static void main(String[] args) {
        Consumer consumer = msg -> System.out::print; // line 1
        consumer.accept("Hello Lambda !");
    }
}
```

This code results in a compilation error.

Which code should be inserted on line 1 for a successful compilation?

- A. Consumer consumer = msg -> { return System.out.print(msg); };
- B. Consumer consumer = var arg > {System.out.print(arg);};
- C. Consumer consumer = (String args) > System.out.print(args);
- D. Consumer consumer = System.out::print;

**Answer:** D

**Explanation:**

```

1  import java.util.*;
2  import java.io.*;
3  import java.nio.file.*;
4  import java.util.List;
5  import java.util.function.Consumer;
6
7  public class Main {
8
9      public static void main(String[] args) {
10         Consumer consumer = System.out::print;
11         consumer.accept("Hello Lambda !");
12     }
13 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

**Result**

CPU Time: 0.16 sec(s), Memory: 32896 kilobyte(s)

Hello Lambda !

**NEW QUESTION 153**

Given:

```

1. public class Main {
2.     public static void greet(String... args) {
3.         System.out.print("Hello ");
4.         for (String arg : args) {
5.             System.out.println(arg);
6.         }
7.     }
8.     public static void main(String[] args) {
9.         Main c = null;
10.        c.greet();
11.    }
12. }

```

What is the result?

- A. NullPointerException is thrown at line 4.
- B. NullPointerException is thrown at line 10.
- C. A compilation error occurs.
- D. Hello

**Answer: D**

**Explanation:**



Console 4 Console 5  
hello  
Completed with exit code: 0

**NEW QUESTION 158**

Given:

```
public interface A {  
    public Iterable a();  
}  
public interface B extends A {  
    public Collection a();  
}  
public interface C extends A {  
    public Path a();  
}  
public interface D extends B, C {  
}
```

Why does D cause a compilation error?

- A. D inherits a() only from C.
- B. D inherits a() from B and C but the return types are incompatible.
- C. D extends more than one interface.
- D. D does not define any method.

**Answer:** B

#### NEW QUESTION 160

Which two statements are true about Java modules? (Choose two.)

- A. Modular jars loaded from --module-path are automatic modules.
- B. Any named module can directly access all classes in an automatic module.
- C. Classes found in -classpath are part of an unnamed module.
- D. Modular jars loaded from -classpath are automatic modules.
- E. If a package is defined in both the named module and the unnamed module, then the package in the unnamed module is ignored.

**Answer:** AC

#### NEW QUESTION 165

Given:

```
package test;  
import java.time.*;  
public class Diary {  
    private LocalDate now = LocalDate.now();  
    public LocalDate getDate() {  
        return now;  
    }  
}
```

and

```
package test;  
public class Tester {  
    public static void main(String[] args) {  
        Diary d = new Diary();  
        System.out.println(d.getDate());  
    }  
}
```

Which statement is true?

- A. Class Tester does not need to import java.time.LocalDate because it is already visible to members of the package test.
- B. All classes from the package java.time
- C. are loaded for the class Diary.
- D. Only LocalDate class from java.time package is loaded.
- E. Tester must import java.time.LocalDate in order to compile.

**Answer:** A

#### NEW QUESTION 167

Given this requirement:

Module vehicle depends on module part and makes its com.vehicle package available for all other modules. Which module-info.java declaration meets the requirement?

A

```
module vehicle{
    requires part;
    exports com.vehicle;
}
```

B

```
module vehicle {
    requires part;
    uses com.vehicle;
}
```

C

```
module vehicle{
    requires part;
    exports com.vehicle to part;
}
```

D

```
module vehicle {
    requires com.vehicle;
    exports part;
}
```

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

**Answer:** A**NEW QUESTION 169**

Given:

```
public method foo() throws FooException {
    ...
}
```

and omitting the throws FooException clause results in a compilation error. Which statement is true about FooException?

- A. FooException is a subclass of RuntimeException.
- B. FooException is unchecked.
- C. The body of foo can only throw FooException.
- D. The body of foo can throw FooException or one of its subclasses.

**Answer:** D**NEW QUESTION 171**

Given:

```
public class Main {
    public static void main(String[] args) {
        Thread t1 = new Thread(new MyThread());
        Thread t2 = new Thread(new MyThread());
        Thread t3 = new Thread(new MyThread());

        t1.start();
        t2.run();
        t3.start();

        t1.start();
    }
}

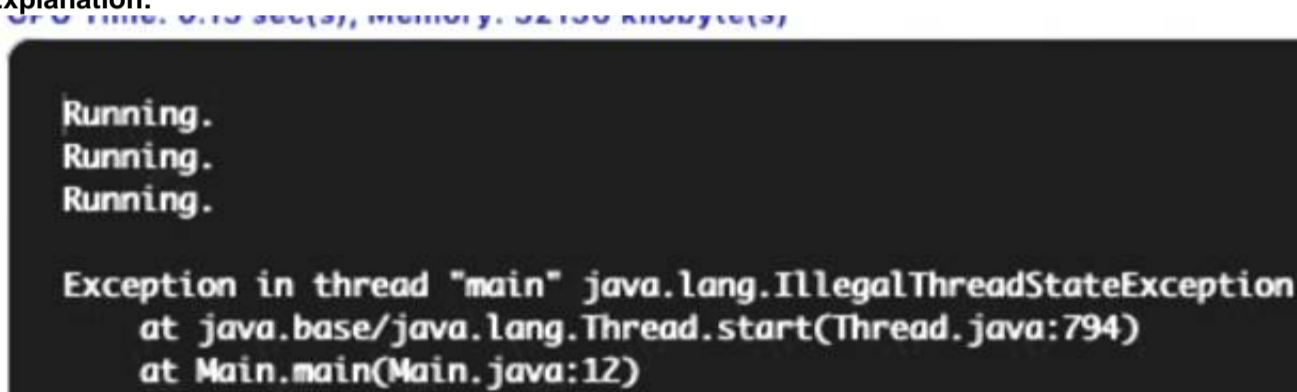
class MyThread implements Runnable {
    public void run() {
        System.out.println("Running.");
    }
}
```

Which one is correct?

- A. An IllegalStateException is thrown at run time.
- B. Three threads are created.
- C. The compilation fails.
- D. Four threads are created.

**Answer: A**

**Explanation:**



#### NEW QUESTION 176

Given:

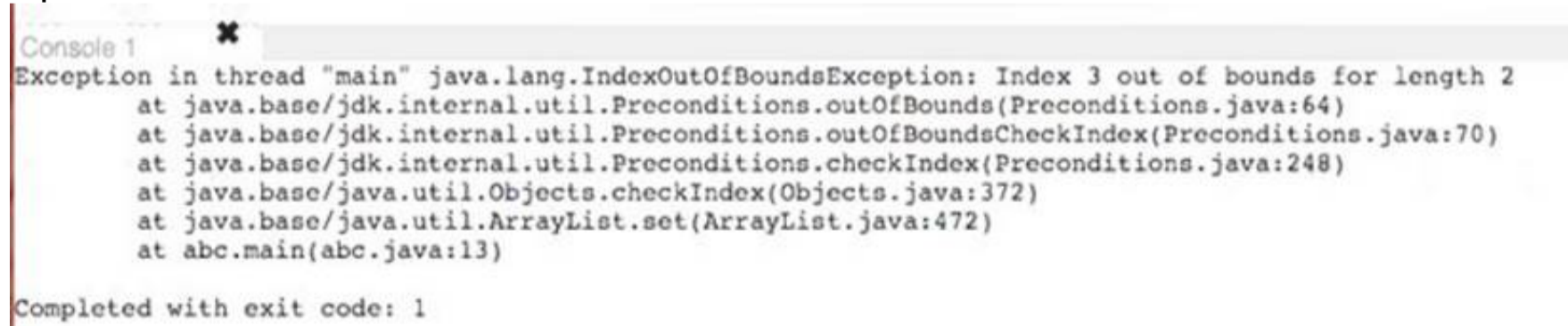
```
var data = new ArrayList<>(); data.add("Peter");
data.add(30); data.add("Market Road"); data.set(1, 25); data.remove(2); data.set(3, 1000L); System.out.print(data);
```

What is the output?

- A. [Market Road, 1000]
- B. [Peter, 30, Market Road]
- C. [Peter, 25, null, 1000]
- D. An exception is thrown at run time.

**Answer: D**

**Explanation:**



#### NEW QUESTION 178

Given:

```
import java.util.List;
import java.util.function.BinaryOperator;
public class Main {
    public static void main(String... args) {
        List<Employee> list = List.of(new Employee("John", 80000.0), new Employee("Scott",
90000.0));
        double starts = 0.0;
        double ratio = 1.0;
        BinaryOperator<Double> bo = (a, b) -> a + b;
double totalSalary = list.stream().map(e -> e.getSalary() * ratio).reduce(starts, bo);
// line 1
        System.out.println("Total salary = " + totalSalary);
    }
}

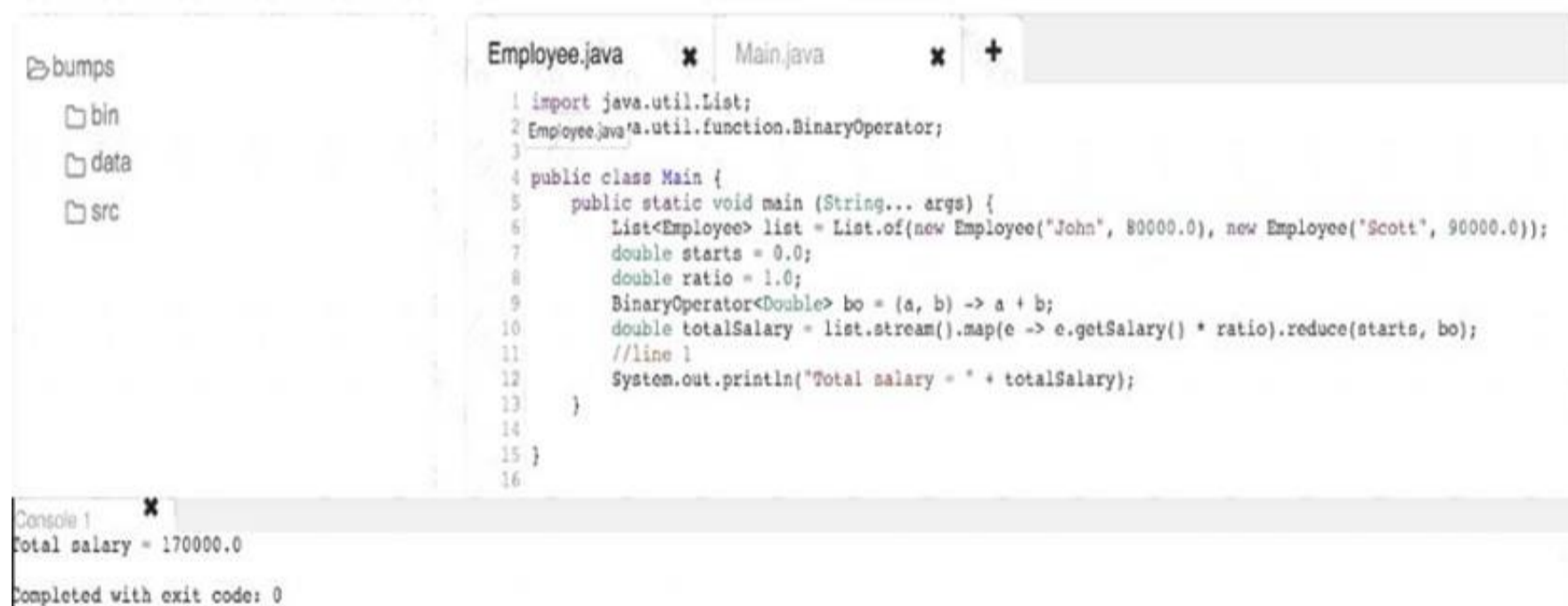
class Employee {
    String name;
    double salary;
    public Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }
    public String getName() { return name; }
    public double getSalary() { return salary; }
}
```

Which statement is equivalent to line 1?

- A. double totalSalary = list.stream().map(e -> e.getSalary() \* ratio).reduce(bo).ifPresent(p -> p.doubleValue());
- B. double totalSalary = list.stream().mapToDouble(e -> e.getSalary() \* ratio).sum;
- C. double totalSalary = list.stream().map(Employee::getSalary \* ratio).reduce(bo).orElse(0.0);
- D. double totalSalary = list.stream().mapToDouble(e -> e.getSalary() \* ratio).reduce(starts, bo);

**Answer: C**

**Explanation:**



The screenshot shows an IDE with two tabs: Employee.java and Main.java. The Main.java tab is active, showing the same code as the question. The console output at the bottom shows "Total salary = 170000.0" and "Completed with exit code: 0".

## NEW QUESTION 181

Given:

```
public class Person {
    private String name;
    public Person(String name) {
        this.name = name;
    }
    public String toString() {
        return name;
    }
}
```

and

```
public class Tester {
    public static void main(String[] args) {
        Person p = null;
        checkPerson(p);
        System.out.println(p);
        p = new Person("Mary");
        checkPerson(p);
        System.out.println(p);
    }
    public static Person checkPerson(Person p) {
        if (p == null) {
            p = new Person("Joe");
        }else{
            p = null;
        }
        return p;
    }
}
```

What is the result?

- A. JoeMarry
- B. Joenull
- C. nullnull
- D. nullMary

**Answer: D**

**Explanation:**



### NEW QUESTION 183

Given:

```
interface MyInterface1 {
    public int method() throws Exception;
    private void pMethod() { /* an implementation of pMethod */ }
}
interface MyInterface2 {
    public static void sMethod() { /* an implementation of sMethod */ }
    public boolean equals();
}
interface MyInterface3 {
    public void method();
    public void method(String str);
}
interface MyInterface4 {
    public void dMethod() { /* an implementation of dMethod */ }
    public void method();
}
interface MyInterface5 {
    public static void sMethod();
    public void method(String str);
}
```

Which two interfaces can be used in lambda expressions? (Choose two.)

- A. MyInterface1
- B. MyInterface3
- C. MyInterface5
- D. MyInterface2
- E. MyInterface4

**Answer:** CD

#### NEW QUESTION 187

Which is a proper JDBC URL?

- A. jdbe.mysql.com://localhost:3306/database
- B. http://localhost.mysql.com:3306/database
- C. http://localhostmysql.jdbc:3306/database
- D. jdbc:mysql://localhost:3306/database

**Answer:** D

#### NEW QUESTION 190

Which statement about access modifiers is correct?

- A. An instance variable can be declared with the static modifier.
- B. A local variable can be declared with the final modifier.
- C. An abstract method can be declared with the private modifier.
- D. An inner class cannot be declared with the public modifier.
- E. An interface can be declared with the protected modifier.

**Answer:** B

#### NEW QUESTION 191

Given:

```
public class Test {  
    private String[] strings;  
}
```

Which two constructors will compile and set the class field strings? (Choose two.)

A.

```
public Test(List<String> strings) {  
    this.strings = strings;  
}
```

B.

```
public Test(String... strings) {  
    strings = strings;  
}
```

C.

```
public Test(String... strings) {  
    this.strings = strings;  
}
```

D.

```
public Test(String strings) {  
    strings = strings;  
}
```

E.

```
public Test(String[] strings) {  
    this.strings = strings;  
}
```

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

Answer: CE

#### NEW QUESTION 195

Given the Person class with age and name along with getter and setter methods, and this code fragment:

```
List<Person> persons = new ArrayList(List.of(new Person(44,"Tom"),
                                             new Person(40,"Aman"),
                                             new Person(40,"Peter")));

persons.sort(Comparator.comparing((Person::getAge))
               .thenComparing(Person::getName)
               .reversed());

persons.forEach(p1->System.out.print(" "+p1.getName()));
```

What will be the result?

- A. Aman Tom Peter
- B. Tom Aman Peter
- C. Aman Peter Tom
- D. Tom Peter Aman

Answer: C

#### NEW QUESTION 196

Given the code fragment:

```
Path source = Paths.get("/repo/a/a.txt"); Path destination = Paths.get("/repo"); Files.move(source, destination); // line 1
Files.delete(source); // line 2
```

Assuming the source file and destination folder exist, what is the result?

- A. A java.nio.file.FileAlreadyExistsException is thrown on line 1.
- B. A java.nio.file.NoSuchFileException is thrown on line 2.
- C. A copy of /repo/a/a.txt is moved to the /repo directory and /repo/a/a.txt is deleted.
- D. a.txt is renamed repo.

Answer: C

#### NEW QUESTION 200

Given:

```
public class Employee {
    private String name;
    private LocalDate birthday;
    // the constructors, getters, and setters methods go here
}
```

and

```
List<Employee> roster = new ArrayList<>();
// ...
Predicate<Employee> y = (Employee e) -> e.getBirthday()
    .isBefore(IsoChronology.INSTANCE.date(1989, 1, 1));
Set<String> s1 = roster.stream()
// Line 1
```

Which code fragment on line 1 makes the s1 set contain the names of all employees born before January 1, 1989?

- A. 

```
.collect(Collectors.partitioningBy(y))
    .get(true)
    .stream()
    .map(Employee::getName)
    .collect(Collectors.toCollection(TreeSet::new));
```
- B. 

```
.collect(Collectors.partitioningBy(y))
    .get(true)
    .map(Employee::getName)
    .collect(Collectors.toSet());
```
- C. 

```
.collect(Collectors.partitioningBy(y, Collectors.mapping(
    Employee::getName, Collectors.toSet())));
```
- D. 

```
.collect(Collectors.partitioningBy(y, Collectors.groupingBy(
    Employee::getName, Collectors.toCollection(TreeSet::new))));
```

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

Answer: B

**NEW QUESTION 204**

Given:

```
enum Color implements Serializable {  
    R(1), G(2), B(3);  
    int c;  
    public Color(int c) {  
        this.c = c;  
    }  
}
```

What action ensures successful compilation?

- A. Replace public Color(int c) with private Color(int c).
- B. Replace int c; with private int c;.
- C. Replace int c; with private final int c;.
- D. Replace enum Color implements Serializable with public enum Color.
- E. Replace enum Color with public enum Color.

**Answer: A****Explanation:**

```
1  
2 import java.io.*;  
3 import java.util.*;  
4 class Hello {  
5  
6  
7     enum Color implements Serializable {  
8         R(1), G(2), B(3);  
9         int c;  
10        private Color (int c) {  
11            this.c = c;  
12        }  
13    }  
14 }
```

**NEW QUESTION 208**

Given:

```
public class Over {  
    public void analyze(Object[] o){  
        System.out.println("I am an object array");  
    }  
    public void analyze(long[] l){  
        System.out.println("I am an array");  
    }  
    public void analyze(Object o){  
        System.out.println("I am an object");  
    }  
    public static void main(String[] args) {  
        int[] nums = new int[10];  
        new Over().analyze(nums); // line 1  
    }  
}
```

What is the output?

- A. I am an object array
- B. The compilation fails due to an error in line 1.
- C. I am an array
- D. I am an object

**Answer: D****NEW QUESTION 210**

Given:

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println(args[0]+args[1]+args[2]);  
    }  
}
```

executed using command:

java Hello "Hello World" Hello World What is the output?

- A. An exception is thrown at runtime.

- B. Hello WorldHello World
- C. Hello World Hello World
- D. Hello WorldHelloWorld
- E. HelloHello WorldHelloWorld

**Answer:** C

#### NEW QUESTION 213

Given:

```
List<Reader> dataFiles = new ArrayList<>();
File indexFile = new File("MyIndex.idx");
try (BufferedReader indexReader =
    new BufferedReader(new FileReader(indexFile))) {
    for(String file = indexReader.readLine(); file != null;
        file = indexReader.readLine()) {
        BufferedReader dataReader = new BufferedReader (
            new FileReader(new File(file))); // Line 1
        dataFiles.add(dataReader); // Line 2
        processData(dataReader); // Line 3
    }
} catch (IOException ex) {
    ...
} finally {
    for(Reader r : dataFiles) {
        try {
            r.close();
        } catch (IOException ex) {
            ...
        } // Line 4
    }
}
```

What will secure this code from a potential Denial of Service condition?

- A. After Line 4, add indexReader.close().
- B. On Line 3, enclose processData(dataReader) with try with resources.
- C. After Line 3, add dataReader.close().
- D. On Line 1, use try with resources when opening each dataReader.
- E. Before Line 1, check the size of dataFiles to make sure it does not exceed a threshold.

**Answer:** B

#### NEW QUESTION 214

Given:

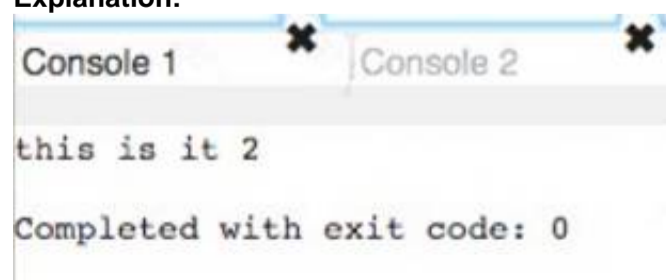
```
public class Tester {
    public static void main(String[] args) {
        String s = "this is it";
        int x = s.indexOf("is");
        s.substring(x+3);
        x = s.indexOf("is");
        System.out.println(s+" "+x);
    }
}
```

What is the result?

- A. is it 1
- B. An IndexOutOfBoundsException is thrown at runtime.
- C. is it 0
- D. this is it 2
- E. this is it 3

**Answer:** D

**Explanation:**



**NEW QUESTION 217**

Given the contents:

MessageBundle.properties file: message=Hello MessageBundle\_en.properties file: message=Hello (en) MessageBundle\_US.properties file: message=Hello (US)

MessageBundle\_en\_US.properties file: message=Hello (en\_US) MessageBundle\_fr\_FR.properties file: message=Bonjour

and the code fragment: Locale.setDefault(Locale.FRANCE);

Locale currentLocale = new Locale.Builder().setLanguage("en").build();

ResourceBundle messages = ResourceBundle.getBundle("MessageBundle", currentLocale); System.out. println(messages.getString("message"));

Which file will display the content on executing the code fragment?

- A. MessageBundle\_en\_US.properties
- B. MessageBundle\_en.properties
- C. MessageBundle\_fr\_FR.properties
- D. MessageBundle\_US.properties
- E. MessageBundle.properties

**Answer: C**

**NEW QUESTION 222**

Which code fragment compiles?

```
A. Comparator comparator = new Comparator<?>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};  
  
B. var comparator = new Comparator<>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};  
  
C. Comparator<> comparator = new Comparator<Integer>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};  
  
D. Comparator<Integer> comparator = new Comparator<>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};
```

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

**Answer: D**

**Explanation:**

```
1 import java.io.*;  
2 import java.util.*;  
3 class abc {  
4     public static void main(String[] args) {  
5  
6         Comparator<Integer> comparator = new Comparator<>() {  
7             public int compare(Integer i, Integer j) {  
8                 return i.compareTo(j);  
9             }  
10        };  
11  
12    }  
13 }|  
14
```

**NEW QUESTION 223**

Given:

```

1. void insertionSort(int values[]) {
2.     int n = values.length;
3.     for (int j = 1; j < n; j++) {
4.         int tmp = values[j];
5.         int i = j - 1;
6.         while ( (i > -1) && (values[i] > tmp) ) {
7.             values[i + 1] = values[i];
8.             i--;
9.         }
10.        values[i + 1] = tmp;
11.    }
12. }

```

After which line can we insert `assert i < 0 || values[i] <= values[i + 1];` to verify that the values array is partially sorted?

- A. after line 8
- B. after line 6
- C. after line 5
- D. after line 10

**Answer: B**

**Explanation:**

```

1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7  import java.util.function.Consumer;
8  import java.util.stream.Stream;
9  import java.util.stream.IntStream;
10
11
12 public class Main {
13
14
15     void insertionSort (int values[]) {
16         int n = values.length;
17         for (int j = 1; j < n; j++) {
18             int tmp = values[j];
19
20             int i = j - 1;
21             assert i < 0 || values[i] <= values[i + 1];
22             while ((i > 1) && (values[i] > tmp) ) {
23                 values[i + 1] = values[i];
24                 i--;
25             }
26             values[i + 1] = tmp;
27
28         }
29     }
30 }
31

```

#### NEW QUESTION 226

Which code is correct?

- A. Runnable r = "Message" > System.out.println();
- B. Runnable r = () > System.out::print;
- C. Runnable r = () -> {System.out.println("Message");};
- D. Runnable r = > System.out.println("Message");
- E. Runnable r = {System.out.println("Message");};

**Answer: C**

#### NEW QUESTION 227

.....

## 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

### 1Z0-819 Practice Exam Features:

- \* 1Z0-819 Questions and Answers Updated Frequently
- \* 1Z0-819 Practice Questions Verified by Expert Senior Certified Staff
- \* 1Z0-819 Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* 1Z0-819 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

**100% Actual & Verified — Instant Download, Please Click**  
**[Order The 1Z0-819 Practice Test Here](#)**