



# Oracle

## Exam Questions 1z0-829

Java SE 17 Developer

## About ExamBible

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

## Found in 1998

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

## Our Advances

### \* 99.9% Uptime

All examinations will be up to date.

### \* 24/7 Quality Support

We will provide service round the clock.

### \* 100% Pass Rate

Our guarantee that you will pass the exam.

### \* Unique Gurantee

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

## NEW QUESTION 1

Given:

```
public class Test {  
    public void sum(int a, int b) {  
        System.out.print(" A");  
    }  
    public void sum(int a, float b) {  
        System.out.print(" B");  
    }  
    public void sum(float a, float b) {  
        System.out.print(" C");  
    }  
    public void sum(double... a) {  
        System.out.print(" D");  
    }  
    public static void main(String[] args) {  
        Test t = new Test();  
        t.sum(10,15.25);  
        t.sum(10, 24);  
        t.sum(10.25,10.25);  
    }  
}
```

What is the result?

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

**Answer:** C

### Explanation:

The answer is C because the code demonstrates the concept of method overloading and type conversion in Java. Method overloading allows different methods to have the same name but different parameters. Type conversion allows values of one data type to be assigned to another data type, either automatically or explicitly. In the code, the class Test has four methods named sum, each with different parameter types: int, float, and double. The main method creates an instance of Test and calls the sum method with different arguments. The compiler will choose the most specific method that matches the arguments, based on the following rules:

? If there is an exact match between the argument types and the parameter types, that method is chosen.

? If there is no exact match, but there is a method with compatible parameter types, that method is chosen. Compatible types are those that can be converted from one to another automatically, such as int to long or float to double.

? If there is more than one method with compatible parameter types, the most specific method is chosen. The most specific method is the one whose parameter types are closest to the argument types in terms of size or precision.

In the code, the following method calls are made:

? test.sum(10, 10.5) -> This matches the sum(int a, float b) method exactly, so it is chosen. The result is 20.5, which is converted to int and printed as 20 (B).

? test.sum(10) -> This does not match any method exactly, but it matches the sum(double a) method with compatible types, as int can be converted to double automatically. The result is 10.0, which is printed as 10 (A).

? test.sum(10.5, 10) -> This does not match any method exactly, but it matches two methods with compatible types: sum(float a, float b) and sum(double a, double b). The latter is more specific, as double is closer to the argument types than float. The result is 20.5, which is printed as 20 (D).

Therefore, the output is B A D. References:

? Oracle Certified Professional: Java SE 17 Developer

? Java SE 17 Developer

? OCP Oracle Certified Professional Java SE 17 Developer Study Guide

? Method Overloading in Java

? Type conversion in Java with Examples

? Java Method Overloading with automatic type conversions

## NEW QUESTION 2

Given:

```
import java.io.Serializable;
public class Software implements Serializable {
    private String title;
    public Software(String title) {
        this.title = title;
        System.out.print("Software ");
    }
    public String toString() { return title; }
}

public class Game extends Software {
    private int players;
    public Game(String title, int players) {
        super(title);
        this.players = players;
        System.out.print("Game ");
    }
    public String toString() { return super.toString()+" "+players; }
}

import java.io.*;
public class AppStore {
    public static void main(String[] args) {
        Software s = new Game("Chess", 2);
        try(ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("game.ser"))) {
            out.writeObject(s);
        } catch (Exception e) {
            System.out.println("write error");
        }
        try(ObjectInputStream in = new ObjectInputStream(new FileInputStream("game.ser"))) {
            s = (Software)in.readObject();
        } catch (Exception e) {
            System.out.println("read error");
        }
        System.out.println(s);
    }
}
```

What is the result?

- A. Software Game Chess 0
- B. Software Game Software Game Chese 2
- C. Software game write error
- D. Software Game Software Game chess 0
- E. Software Game Chess 2
- F. Software Game read error

**Answer:** B

### Explanation:

The answer is B because the code uses the writeObject and readObject methods of the ObjectOutputStream and ObjectInputStream classes to serialize and deserialize the Game object. These methods use the default serialization mechanism, which writes and reads the state of the object's fields, including the inherited ones. Therefore, the title field of the Software class is also serialized and deserialized along with the players field of the Game class. The toString method of the Game class calls the toString method of the Software class using super.toString(), which returns the value of the title field. Hence, when the deserialized object is printed, it shows ??Software Game Software Game Chess 2??.

References:

- ? Oracle Certified Professional: Java SE 17 Developer
- ? Java SE 17 Developer
- ? OCP Oracle Certified Professional Java SE 17 Developer Study Guide
- ? Serialization and Deserialization in Java with Example

## NEW QUESTION 3

Which statement is true?

- A. The tryLock () method returns a boolean indicator immediately regardless if it has or has not managed to acquire the lock.
- B. The tryLock () method returns a boolean indicator immediately if it has managed to acquire the lock, otherwise it waits for the lock acquisition.
- C. The lock () method returns a boolean indicator immediately if it has managed to acquire the lock, otherwise it waits for the lock acquisition.
- D. The Lock () method returns a boolean indicator immediately regardless if it has or has not managed to acquire the lock

**Answer:** A

**Explanation:**

The tryLock () method of the Lock interface is a non-blocking attempt to acquire a lock. It returns true if the lock is available and acquired by the current thread, and false otherwise. It does not wait for the lock to be released by another thread. This is different from the lock () method, which blocks the current thread until the lock is acquired, and does not return any value. References: [https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/concurrent/locks/Lock.html#tryLock\(\)](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/concurrent/locks/Lock.html#tryLock()), 3, 4, 5

**NEW QUESTION 4**

Given the code fragment:

```
String myStr = "Hello Java 17";
String myTextBlk1 = ""
                "Hello Java 17""";
String myTextBlk2 = ""
                "Hello Java 17
                """;

System.out.print(myStr.equals(myTextBlk1)+":");
System.out.print(myStr.equals(myTextBlk2)+":");
System.out.print(myTextBlk1.equals(myTextBlk2)+":");
System.out.println(myTextBlk1.intern() == myTextBlk2.intern());
```

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

**Answer:** C

**Explanation:**

The code fragment compares four pairs of strings using the equals() and intern() methods. The equals() method compares the content of two strings, while the intern() method returns a canonical representation of a string, which means that it returns a reference to an existing string with the same content in the string pool. The string pool is a memory area where strings are stored and reused to save space and improve performance. The results of the comparisons are as follows:  
? s1.equals(s2): This returns true because both s1 and s2 have the same content, ??Hello Java 17??.  
? s1 == s2: This returns false because s1 and s2 are different objects with different references, even though they have the same content. The == operator compares the references of two objects, not their content.  
? s1.intern() == s2.intern(): This returns true because both s1.intern() and s2.intern() return a reference to the same string object in the string pool, which has the content ??Hello Java 17??. The intern() method ensures that there is only one copy of each distinct string value in the string pool.  
? ??Hello Java 17?? == s2: This returns false because ??Hello Java 17?? is a string literal, which is automatically interned and stored in the string pool, while s2 is a string object created with the new operator, which is not interned by default and stored in the heap. Therefore, they have different references and are not equal using the == operator.  
References: String (Java SE 17 & JDK 17) - Oracle

**NEW QUESTION 5**

Given the code fragment:

```
Stream<String> s1 = Stream.of("A", "B", "C", "B");
Stream<String> s2 = Stream.of("A", "D", "E");
Stream.concat(s1, s2).parallel().distinct().forEach(element -> System.out.print(element));
```

What is the result:

- A. ADEACB // the order of element is unpredictable
- B. ABCE
- C. ABCDE // the order of elements is unpredictable
- D. ABBCDE // the order of elements is unpredictable

**Answer:** D

**Explanation:**

The answer is D because the code fragment uses the Stream API to create two streams, s1 and s2, and then concatenates them using the concat() method. The resulting stream is then processed in parallel using the parallel() method, and the distinct() method is used to remove duplicate elements. Finally, the forEach() method is used to print the elements of the resulting stream to the console. Since the order of elements in a parallel stream is unpredictable, the output could be any of the options given, but option D is the most likely. References:  
? Oracle Certified Professional: Java SE 17 Developer  
? Java SE 17 Developer  
? OCP Oracle Certified Professional Java SE 17 Developer Study Guide  
? Parallelizing Streams



#### NEW QUESTION 6

Which statement is true about migration?

- A. Every module is moved to the module path in a top-down migration.
- B. Every module is moved to the module path in a bottom-up migration.
- C. The required modules migrate before the modules that depend on them in a top-down migration.
- D. Unnamed modules are automatic modules in a top-down migration.

**Answer: B**

#### Explanation:

The answer is B because a bottom-up migration is a strategy for modularizing an existing application by moving its dependencies to the module path one by one, starting from the lowest-level libraries and ending with the application itself. This way, each module can declare its dependencies on other modules using the module-info.java file, and benefit from the features of the Java Platform Module System (JPMS), such as reliable configuration, strong encapsulation, and service loading.

Option A is incorrect because a top-down migration is a strategy for modularizing an existing application by moving it to the module path first, along with its dependencies as automatic modules. Automatic modules are non-modular JAR files that are treated as modules with some limitations, such as not having a module descriptor or a fixed name. A top-down migration allows the application to use the module path without requiring all of its dependencies to be modularized first.

Option C is incorrect because a top-down migration does not require any specific order of migrating modules, as long as the application is moved first and its dependencies are moved as automatic modules. A bottom-up migration, on the other hand, requires the required modules to migrate before the modules that depend on them.

Option D is incorrect because unnamed modules are not automatic modules in any migration strategy. Unnamed modules are modules that do not have a name or a module descriptor, such as classes loaded from the class path or dynamically generated classes. Unnamed modules have unrestricted access to all other modules, but they cannot be accessed by named modules, except through reflection with reduced security checks. References:

? Oracle Certified Professional: Java SE 17 Developer

? Java SE 17 Developer

? OCP Oracle Certified Professional Java SE 17 Developer Study Guide

? Migrating to Modules (How and When) - JavaDeploy

? Java 9 Modularity: Patterns and Practices for Developing Maintainable Applications

#### NEW QUESTION 7

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()) {
            f.open();
        }
    }
}
```

Which two modifications enable the code to print Open Close?

A)

At line n2, insert:

```
final void close() {  
    System.out.print("Close ");
```

B)

Replace line n1 with:

```
class Folder extends Closeable {
```

C)

Replace line n1 with:

```
class Folder extends Exception {
```

D)

Replace line n1 with:

```
class Folder implements AutoCloseable {
```

E)

At line n2, insert:

```
public void close() throws IOException {  
    System.out.print("Close ");  
}
```

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

**Answer:** BE

**Explanation:**

The code given is a try-with-resources statement that declares a resource of type AutoCloseable. The resource is an anonymous class that implements the AutoCloseable interface and overrides the close() method. The code also has a print() method that prints the value of the variable s. The code is supposed to print ??Open Close??, but it does not compile because of two errors.

The first error is at line n1, where the anonymous class is missing a semicolon at the end of its declaration. This causes a syntax error and prevents the code from compiling. To fix this error, option B adds a semicolon after the closing curly brace of the anonymous class.

The second error is at line n2, where the print() method is called without an object reference. This causes a compilation error because the print() method is not static and cannot be invoked without an object. To fix this error, option E adds an object reference to the print() method by using the variable t.

Therefore, options B and E are correct and enable the code to print ??Open Close??.

#### NEW QUESTION 8

Given:

```
1. class Item {
2.     String name;
3.     public static void display() {
4.         name = "Vase";
5.         System.out.println(name);
6.     }
7.     public void display(String design) {
8.         this.name += name;
9.         System.out.println(name);
10.    }
11. }
12. public class App {
13.     public static void main(String[] args) {
14.         Item i1 = new Item();
15.         i1.display("Flower");
16.     }
17. }
```

Which action enables the code to compile?

- A. Replace 15 with item.display ("Flower");
- B. Replace 2 with static string name;
- C. Replace 7 with public void display (string design) {
- D. Replace 3 with private static void display () {

**Answer: C**

#### Explanation:

The answer is C because the code fragment contains a syntax error in line 7, where the method display is declared without any parameter type. This causes a compilation error, as Java requires the parameter type to be specified for each method parameter. To fix this error, the parameter type should be added before the parameter name, such as string design. This will enable the code to compile and run without any errors. References:

? Oracle Certified Professional: Java SE 17 Developer

? Java SE 17 Developer

? OCP Oracle Certified Professional Java SE 17 Developer Study Guide

? Java Methods

#### NEW QUESTION 9

Given the code fragment:



```
int a = 2;
int b = ~a;
int c = a^b;
boolean d = a < b & a > c++;
System.out.println(d + " " + c);
boolean e = a > b && a > c++;
System.out.println(e + " " + c);
```

What is the result?

- A. false 1false 2
- B. true 1false 2
- C. false 1ture 2
- D. falase 0true 1

**Answer:** B

**Explanation:**

The code fragment is comparing the values of a, b, and c using the < and > operators. The first comparison, d, is checking if a is less than b and greater than c. Since a is equal to 2, b is equal to -2, and c is equal to -4, this comparison will evaluate to true. The second comparison, e, is checking if a is greater than b and a is greater than c. Since a is equal to 2, b is equal to -2, and c is equal to -4, this comparison will evaluate to false. Therefore, the result will be true 1 false 2. References: Operators (The Java™ Tutorials > Learning the Java Language - Oracle)

**NEW QUESTION 10**

Given:

```
package com.transport.vehicle.cars;

public interface Car {
    int getSpeed();
}

and

package com.transport.vehicle.cars.impl;

import com.transport.vehicle.cars.Car;

public class CarImpl implements Car {
    private int speed;

    public CarImpl() {
        this(10);
    }

    public CarImpl (int speed) {
        this.speed = speed;
    }

    @Override
    public int getSpeed() {
        return speed;
    }
}
```

Which two should the module-info file include for it to represent the service provider interface?

- A. Requires cm.transport.vehicle,cars:
- B. Provides com.transport.vehicle.cars.Car with com.transport.vehicle.car
- C. impt, CatImpl;
- D. Requires cm.transport.vehicle,cars:
- E. Provides com.transport.vehicle.cars.Car impl,CarImpl to com.transport.vehicle.car
- F. Cars
- G. exports com.transport.vehicle.cars.Car;
- H. Exports com.transport.vehicle.cars;
- I. Exports com.transport.vehicle;

**Answer:** BE

**Explanation:**

The answer is B and E because the module-info file should include a provides directive and an exports directive to represent the service provider interface. The provides directive declares that the module provides an implementation of a service interface, which is com.transport.vehicle.cars.Car in this case. The with clause specifies the fully qualified name of the service provider class, which is com.transport.vehicle.cars.impl.CarImpl in this case. The exports directive declares that the module exports a package, which is com.transport.vehicle.cars in this

case, to make it available to other modules. The package contains the service interface that other modules can use.

Option A is incorrect because requires is not the correct keyword to declare a service provider interface. Requires declares that the module depends on another module, which is not the case here.

Option C is incorrect because it has a typo in the module name. It should be com.transport.vehicle.cars, not cm.transport.vehicle.cars.

Option D is incorrect because it has a typo in the keyword provides. It should be provides, not Provides. It also has a typo in the service interface name. It should be com.transport.vehicle.cars.Car, not com.transport.vehicle.cars.Car impl. It also has an unnecessary to clause, which is used to limit the accessibility of an exported package to specific modules.

Option F is incorrect because it exports the wrong package. It should export com.transport.vehicle.cars, not com.transport.vehicle.cars.impl. The impl package contains the service provider class, which should not be exposed to other modules.

Option G is incorrect because it exports the wrong package. It should export com.transport.vehicle.cars, not com.transport.vehicle. The vehicle package does not contain the service interface or the service provider class. References:

? Oracle Certified Professional: Java SE 17 Developer

? Java SE 17 Developer

? OCP Oracle Certified Professional Java SE 17 Developer Study Guide

? Java Modules - Service Interface Module - GeeksforGeeks

? Java Service Provider Interface | Baeldung

#### NEW QUESTION 10

Given the code fragment:

```
// line n1
String input = console.readLine("Input a number: ");
int number = Integer.parseInt(input);

if (number % 2 == 0) {
    System.out.println(number + " is even.");
} else {
    System.out.println(number + " is odd");
}
```

Which code line n1, obtains the java.io.Console object?

A)

```
Console console = System.console(System.in);
```

B)

```
Console console = Console.getInstance();
```

C)

```
Console console = System.console();
```

D)

```
Console console = new Console(System.in);
```

E)

```
Console console = new Console(new InputStreamReader(System.in));
```

A. Option A

B. Option B

- C. Option C
- D. Option D
- E. Option E

**Answer:** A

**Explanation:**

The code fragment is trying to obtain the `java.io.Console` object, which is a class that provides methods to access the character-based console device, if any, associated with the current Java virtual machine. The correct way to obtain the Console object is to call the static method `Console console()` in the `java.lang.System` class. This method returns the unique Console object associated with the current Java virtual machine, if any. Therefore, option A is correct, as it calls `System.console()` and assigns it to a Console variable. References:

- ? <https://docs.oracle.com/javase/17/docs/api/java.base/java/io/Console.html>
- ? [https://docs.oracle.com/javase/17/docs/api/java.base/java/lang/System.html#console\(\)](https://docs.oracle.com/javase/17/docs/api/java.base/java/lang/System.html#console())
- ? [https://education.oracle.com/products/trackp\\_OCPJSE17](https://education.oracle.com/products/trackp_OCPJSE17)
- ? <https://mylearn.oracle.com/ou/learning-path/java-se-17-developer/99487>

**NEW QUESTION 13**

Given the content of the `in.txt` file: 23456789  
and the code fragment:

```
char[] buffer = new char[8];
int count = 0;
try(FileReader in = new FileReader("in.txt");
    FileWriter out = new FileWriter("out.txt")) {
    while((count = in.read(buffer)) != -1) {
        out.write(buffer);
    }
}
```

What is the content of the `out.txt` file?

- A. 01234567801234
- B. 012345678
- C. 0123456789234567
- D. 0123456789
- E. 012345678901234
- F. 01234567

**Answer:** D

**Explanation:**

The answer is D because the code fragment reads the content of the `in.txt` file and writes it to the `out.txt` file. The content of the `in.txt` file is `??23456789??`. The code fragment uses a char array buffer of size 8 to read the content of the `in.txt` file. The while loop reads the content of the `in.txt` file and writes it to the `out.txt` file until the end of the file is reached. Therefore, the content of the `out.txt` file will be `??0123456789??`.

**NEW QUESTION 16**

Given:



```
public class Test {  
    public static void main(String[] args) {  
        final int x = 2;  
        int y = x;  
        while (y<3) {  
            switch (y) {  
                case 0+x:  
                    y++;  
                case 1:  
                    y++;  
            }  
        }  
        System.out.println(y);  
    }  
}
```

What is the result?

- A. 4
- B. 2
- C. 6
- D. Nothing is printed because of an indefinite loop.
- E. Compilation fails.
- F. 5
- G. A runtime exception is thrown.
- H. 3

**Answer:** E

**Explanation:**

The code will not compile because the variable `x` is declared as final and then it is being modified in the switch statement. This is not allowed in Java. A final variable is a variable whose value cannot be changed once it is initialized<sup>1</sup>. The switch statement tries to assign different values to `x` depending on the value of `y`, which violates the final modifier. The compiler will report an error: The final local variable x cannot be assigned. It must be blank and not using a compound assignment. References: The final Keyword (The Java™ Tutorials > Learning the Java Language > Classes and Objects)

**NEW QUESTION 18**

Given:

Captions.properties file:

```
user = UserName
```

Captions\_en.properties file:

```
user = User name (EN)
```

Captions\_US.properties file:

```
message = User name (US)
```

Captions\_en\_US.properties file:

```
message = User name (EN - US)
```

and the code fragment:

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

ResourceBundle captions = ResourceBundle.getBundle("Captions.properties", currentLocale);
System.out.println(captions.getString("user"));
```

What is the result?

- A. User name (US)
- B. The program throws a MissingResourceException.
- C. User name (EN – US)
- D. UserName
- E. User name (EN)

**Answer: B**

**Explanation:**

The answer is B because the code fragment contains a logical error that causes a MissingResourceException at runtime. The code fragment tries to load a resource bundle with the base name ??Captions.properties?? and the locale ??en\_US??. However, there is no such resource bundle available in the classpath. The available resource bundles are:

- ? Captions.properties
- ? Captions\_en.properties
- ? Captions\_US.properties
- ? Captions\_en\_US.properties

The ResourceBundle class follows a fallback mechanism to find the best matching resource bundle for a given locale. It first tries to find the resource bundle with the exact locale, then it tries to find the resource bundle with the same language and script, then it tries to find the resource bundle with the same language, and finally it tries to find the default resource bundle with no locale. If none of these resource bundles are found, it throws a MissingResourceException.

In this case, the code fragment is looking for a resource bundle with the base name ??Captions.properties?? and the locale ??en\_US??. The ResourceBundle class will try to find the following resource bundles in order:

- ? Captions.properties\_en\_US
- ? Captions.properties\_en
- ? Captions.properties

However, none of these resource bundles exist in the classpath. Therefore, the ResourceBundle class will throw a MissingResourceException.

To fix this error, the code fragment should use the correct base name of the resource bundle family, which is ??Captions?? without the ??properties?? extension. For example: ResourceBundle captions = ResourceBundle.getBundle(??Captions??, currentLocale); This will load the appropriate resource bundle for the current locale, which is ??Captions\_en\_US.properties?? in this case. References:

- ? Oracle Certified Professional: Java SE 17 Developer
- ? Java SE 17 Developer
- ? OCP Oracle Certified Professional Java SE 17 Developer Study Guide
- ? ResourceBundle (Java Platform SE 8 )
- ? About the ResourceBundle Class (The Java™ Tutorials > Internationalization)

**NEW QUESTION 19**

Assume you have an automatic module from the module path display-ascii-0.2. jar. Which name is given to the automatic module based on the given JAR file?

- A. Display.ascii
- B. Display-ascii-0.2
- C. Display-ascii
- D. Display-ascii-0

**Answer: C**

**Explanation:**

An automatic module name is derived from the name of the JAR file when it does not contain a module-info.class file. If the JAR file has an ??Automatic-Module-Name?? attribute in its main manifest, then its value is the module name. Otherwise, the module name is derived from the JAR file??s name by removing any version numbers and converting it to lower case. Therefore, for a JAR named display-ascii-0.2.jar, the automatic module name would be display-ascii, following these rules.

#### NEW QUESTION 24

Given the directory structure:

```
module1:
    p1\
        Doc.java
    p2\
        Util.java
```

Given the definition of the Doc class:

```
package p1;
    public sealed class Doc permits WordDoc {
}
```

Which two are valid definition of the wordDoc class?

- A. Package p1;Public non-sealed class wordDoc extends Doc ()
- B. Package p1;Public class wordDoc extends Doc ()
- C. Package p1, p2;Public non-sealed class WordDoc extends Doc ()
- D. Package p1, p2;Public sealed class WordDoc extends Doc ()
- E. Package p1,non-sealed abstract class WordDoc extends Doc ()
- F. Package p1;Public final class WordDoc extends Doc ()

**Answer:** AF

#### **Explanation:**

The correct answer is A and F because the wordDoc class must be a non-sealed class or a final class to extend the sealed Doc class. Option B is incorrect because the wordDoc class must be non-sealed or final. Option C is incorrect because the wordDoc class cannot be in a different package than the Doc class. Option D is incorrect because the wordDoc class cannot be a sealed class. Option E is incorrect because the wordDoc class cannot be an abstract class.  
References: Oracle Certified Professional: Java SE 17 Developer, 3 Sealed Classes - Oracle Help Center

#### NEW QUESTION 27

Given the code fragments:

```
class Test {
    volatile int x = 1;
    AtomicInteger xObj = new AtomicInteger(1);
}

and

public static void main(String[] args) {
    Test t = new Test();
    Runnable r1 = () -> {
        Thread trd = Thread.currentThread();
        while (t.x < 3 ) {
            System.out.print(trd.getName()+" : "+t.x+" : ");
            t.x++;
        }
    };
    Runnable r2 = () -> {
        Thread trd = Thread.currentThread();
        while (t.xObj.get() < 3) {
            System.out.print(trd.getName()+" : "+t.xObj.get()+" : ");
            t.xObj.getAndIncrement();
        }
    };
    Thread t1 = new Thread(r1,"t1");
    Thread t2 = new Thread(r2,"t2");
    t1.start();
    t2.start();
}
```

Which is true?

- A. The program prints t1 : 1: t2 : 1: t1 : t2 : 2 : in random order.
- B. The program prints t1 : 1 : t2: 1 : t1 : 2 : t2: 2:
- C. The program prints t1 : 1: t2 : 1: t1 : 1 : t2 : 1 : indefinitely
- D. The program prints an exception

**Answer:** B

**Explanation:**

The code creates two threads, t1 and t2, and starts them. The threads will print their names and the value of the Atomic Integer object, x, which is initially set to 1. The threads will then increment the value of x and print their names and the new value of x. Since the threads are started at the same time, the output will be in random order.

However, the final output will always be t1 : 1 : t2: 1 : t1 : 2 : t2: 2: References: AtomicInteger (Java SE 17 & JDK 17) - Oracle

**NEW QUESTION 30**

.....



## Relate Links

**100% Pass Your 1z0-829 Exam with ExamBible Prep Materials**

<https://www.exambible.com/1z0-829-exam/>

## Contact us

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

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