

# Exam Questions 1z0-829

Java SE 17 Developer

<https://www.2passeasy.com/dumps/1z0-829/>



## NEW QUESTION 1

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 Chess 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 2

Given the code fragment:

```
List lst = new ArrayList();
lst.add("e1");
lst.add("e3");
lst.add("e2");

int x1 = Collections.binarySearch(lst, "e3");
System.out.println(x1);
Collections.sort(lst);
int x2 = Collections.binarySearch(lst, "e3");
System.out.println(x2);

Collections.reverse(lst);
int x3 = Collections.binarySearch(lst, "e3");
System.out.println(x3);
```

What is the result?

- A. 2
- B. -2
- C. 22E.111F.12-4

**Answer:** B

**Explanation:**

The code fragment uses the Collections.binarySearch method to search for the string "e3" in the list. The first search returns the index of the element, which is 2. The second search returns the index of the element, which is 0. The third search returns the index of the element, which is -4. The final result is 2. References: Collections (Java SE 17 & JDK 17) - Oracle

**NEW QUESTION 3**

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 4**

Assuming that the data, txt file exists and has the following content:

Text1 Text2 Text3

Given the code fragment:

```
try {  
    Path p = new File("data.txt").toPath();  
    Stream lines = Files.lines(p);  
    String data = lines.collect(Collectors.joining("-"));  
    System.out.println(data);  
    String data2 = Files.readAllLines(p).get(3);  
    System.out.println(data2);  
} catch (IOException ex) {  
    System.out.println(ex);  
}
```

What is the result?

- A. text1- text2- text3- text3
- B. text1-text2-text3 text1text2 text3
- C. text1-text2-text3A java.lang.indexoutofBoundsException is thrown.
- D. text1-text2-text3 text3

**Answer:** D

**Explanation:**

The answer is D because the code fragment reads the file ??data.txt?? and collects all the lines in the file into a single string, separated by hyphens. Then, it prints the resulting string. Next, it attempts to read the fourth line in the file (index 3) and print it. However, since the file only has three lines, an

IndexOutOfBoundsException is thrown. References:

? Oracle Certified Professional: Java SE 17 Developer

? Java SE 17 Developer

? OCP Oracle Certified Professional Java SE 17 Developer Study Guide

? Read contents of a file using Files class in Java

**NEW QUESTION 5**

Which two code fragments compile?

A)

```
class L6 {  
    public static void main(String[] args) {  
        var x = new ArrayList<>();  
        x.add(10);  
        x.add("30");  
        System.out.println(x);  
    }  
}
```

B)

```
class L2 {  
    public void m(int x) {  
        var x = 10;  
    }  
}
```

C)

```
class A {}  
class B extends A {}  
class L4 {  
    public static void main(String[] args) {  
        var x = new A();  
        x = new B();  
    }  
}
```

D)

```
class L3 {  
    public static void main(String[] args) {  
        var a = 10;  
        a = "30";  
    }  
}
```

E)

```
class L5 {  
    public void m() {  
        var strVar = null;  
    }  
}
```

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

**Answer:** BE

**Explanation:**

The two code fragments that compile are B and E. These are the only ones that use the correct syntax for declaring and initializing a var variable. The var keyword is a reserved type name that allows the compiler to infer the type of the variable based on the initializer expression. However, the var variable must have an initializer, and the initializer must not be null or a lambda expression. Therefore, option A is invalid because it does not have an initializer, option C is invalid because it has a null initializer, and option D is invalid because it has a lambda expression as an initializer. Option B is valid because it has a String initializer, and option E is valid because it has an int initializer. <https://docs.oracle.com/en/java/javase/17/language/local-variable-type-inference.html>

## NEW QUESTION 6

Given:

```
public class Test {
    public static void main(String[] args) {
        List<String> elements =
            Arrays.asList("car", "truck", "car",
                "bicycle", "car", "truck", "motorcycle");
        Map<String, Long> outcome =
            elements.stream().collect(Collectors.groupingBy(Function.identity(), Collectors.counting() ) );
        System.out.println(outcome);
    }
}
```

What is the result?

- A. Bicycle =7, car=7, motorcycle=7, truck=7)
- B. (3:bicycle, 0:car, 0:motorcycle, 5:truck)
- C. (Bicycle, car, motorcycle, truck)
- D. Bicycle-1, car=3, motorcycle=1, truck=2)
- E. Compilation fails.

**Answer:** E

### Explanation:

The answer is E because the code fragment contains several syntax errors that prevent it from compiling. Some of the errors are:

? The enum declaration is missing a semicolon after the list of constants.

? The enum constants are not capitalized, which violates the Java naming convention for enums.

? The switch expression is missing parentheses around the variable name.

? The case labels are missing colons after the enum constants.

? The default label is missing a break statement, which causes a fall-through to the next case.

? The println statement is missing a closing parenthesis and a semicolon. A possible corrected version of the code fragment is:

```
enum Vehicle { BICYCLE, CAR, MOTORCYCLE, TRUCK; } public class Test { public static void main(String[] args) { Vehicle v = Vehicle.BICYCLE; switch (v) {
case BICYCLE:
```

```
System.out.print(??1??); break; case CAR: System.out.print(??3??); break; case MOTORCYCLE: System.out.print(??1??); break; case TRUCK:
```

```
System.out.print(??2??); break; default: System.out.print(??0??); break; } System.out.println(); } }
```

This would print 1 as the output. References:

? Oracle Certified Professional: Java SE 17 Developer

? Java SE 17 Developer

? OCP Oracle Certified Professional Java SE 17 Developer Study Guide

? Enum Types

? The switch Statement

## NEW QUESTION 7

Given:

```
public class Test {
    static interface Animal {
    }

    static class Dog implements Animal {
    }

    private static void play(Animal a) {
        System.out.print("flips");
    }

    private static void play(Dog d) {
        System.out.print("runs");
    }

    public static void main(String[] args) {
        Animal a1 = new Dog();
        Dog a2 = new Dog();
        play(a1);
        play(a2);
    }
}
```

What is the result?

- A. flipsflips
- B. Compilation fails
- C. flipsruns
- D. runsflips
- E. runsruns

**Answer:** B

**Explanation:**

The code fragment will fail to compile because the play method in the Dog class is declared as private, which means that it cannot be accessed from outside the class. The main method is trying to call the play method on a Dog object, which is not allowed. Therefore, the code fragment will produce a compilation error.

**NEW QUESTION 8**

Given:

```
interface IFace {
    public void m1();
    public default void m2() {
        System.out.println("m2");
    }
    public static void m3() {
        System.out.println("m3");
    }
    private void m4() {
        System.out.println("m4");
    }
}

class MyC implements IFace {
    public void m1() {
        System.out.println("Hello");
    }
}
```

Which two method invocation execute?

- A. IFace myclassobj = new Myc (); myclassObj.m3 ();
- B. Ifnce.m3 ();
- C. iFace mucloassObj = new Myc (); myClassObj.m4();
- D. new MyC() .m2 ();
- E. IFace .,4():
- F. IFace.m2();

**Answer:** DE

**Explanation:**

The code given is an interface and a class that implements the interface. The interface has three methods, m1(), m2(), and m3(). The class has one method, m1(). The only two method invocations that will execute are D and E. D is a call to the m2() method in the class, and E is a call to the m3() method in the interface.

References: [https://education.oracle.com/products/trackp\\_OCPJSE17](https://education.oracle.com/products/trackp_OCPJSE17), 3, 4, 5

**NEW QUESTION 9**

Which statement is true about modules?

- A. Automatic and unnamed modules are on the module path.
- B. Only unnamed modules are on the module path.
- C. Automatic and named modules are on the module path.
- D. Only named modules are on the module path.
- E. Only automatic modules are on the module path.

**Answer:** C

**Explanation:**

A module path is a sequence of directories that contain modules or JAR files. A named module is a module that has a name and a module descriptor (module-info.class) that declares its dependencies and exports. An automatic module is a module that does not have a module descriptor, but is derived from the name and contents of a JAR file. Both named and automatic modules can be placed on the module path, and they can be resolved by the Java runtime. An unnamed module is a special module that contains all the classes that are not in any other module, such as those on the class path. An unnamed module is not on the module path, but it can read all other modules.

**NEW QUESTION 10**

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 10**

Given the code fragment:

```
Pet p = new Pet("Dog");
Pet p1 = p;
p1.name = "Cat";
p = p1;
System.out.println(p.name);
p = null;
System.out.println(p1.name);
```

What is the result?

- A. A.Cat Dog
- B. A NullPointerException is thrown CatCat
- C. Dog Dog
- D. Cat null

**Answer:** D**Explanation:**

The answer is E because the code fragment creates a new Pet object with the name ??Dog?? and assigns it to the variable p. Then, it assigns p to p1. Next, it changes the name of p1 to ??Cat??. Then, it assigns p1 to p. Finally, it sets p to null and prints the name of p and p1. The output will be ??Cat?? and ??null?? because p is set to null and p1 still points to the Pet object with the name ??Cat??.

**NEW QUESTION 15**

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 19**

Given the code fragment:

```
String s = "10_00";
Integer s2 = 10_00;
// Line n1
System.out.println(res);
```

Which two statements at Line n1 independently enable you to print 1250?

- A. Integer res = 250 + integer.parseInt (s)
- B. Integer res = 250 + s;
- C. Integer res = 250 + integer (s2):
- D. Integer res= 250 + s2;
- E. Integer res = 250 + integer . valueOf (s);
- F. Integer res = 250; Res = + s2;

**Answer:** AE

**Explanation:**

The code fragment is creating a string variable `s` with the value `"10_00"` and an integer variable `s2` with the value 10. The string `s` is using an underscore as a separator for readability, which is allowed in Java SE 17.1. The question is asking for two statements that can add 250 to the numeric value of `s` and assign it to an integer variable `res`. The correct answers are A and E because they use the methods `parseInt` and `valueOf` of the `Integer` class to convert the string `s` to an integer. Both methods interpret the string as a signed decimal integer and return the equivalent `int` or `Integer` value. The other options are incorrect because they either use invalid syntax, such as B and C, or they do not convert the string `s` to an integer, such as D and F. References: Binary Literals (The Java™ Tutorials > Learning the Java Language > Numbers and Strings), Integer (Java SE 17 & JDK 17), Integer (Java SE 17 & JDK 17)

**NEW QUESTION 24**

Given:

```
public class App{
    String name;
    public App(String name){
        this.name = name;
    }
    public static void main(String args[]) {
        App t1= new App("t1");
        App t2= new App("t2");
        t1 = t2;
        t1 = null;
        System.out.println("GC");
    }
}
```

Which statement is true while the program prints GC?

- A. Only the object referenced by t2 is eligible for garbage collection.
- B. Both the objects previously referenced by t1 are eligible for garbage collection.
- C. None of the objects are eligible for garbage collection.
- D. Only one of the objects previously referenced by t1 is eligible for garbage collection.

**Answer:** B

#### NEW QUESTION 25

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 28

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 33**

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 37**

Given the code fragment:

```
Integer rank = 4;
switch (rank) {
    case 1,4 -> System.out.println("Range1");
    case 5,8 -> System.out.println("Range2");
    case 9,10 -> System.out.println("Range3");
    default -> System.out.println("Not a valid rank.");
}
```

What is the result?

- A. Range 1Range 2Range 3
- B. Range1Note a valid rank.
- C. Range 1Range 2Range 3Range 1Not a valida rank
- D. Range 1

**Answer:** C

**Explanation:**

The code fragment is using the switch statement with the new Java 17 syntax. The switch statement checks the value of the variable rank and executes the corresponding case statement. In this case, the value of rank is 4, so the first case statement is executed, printing ??Range1??. The second and third case statements are also executed, printing ??Range2?? and ??Range3??. The default case statement is also executed, printing ??Not a valid rank??. References: Java Language Changes - Oracle Help Center

**NEW QUESTION 41**

Daylight Saving Time (DST) is the practice of advancing clocks at the start of spring by one hour and adjusting them backward by one hour in autumn.

Considering that in 2021, DST in Chicago (Illinois) ended on November 7th at 2 AM, and given the fragment:

```
ZoneId zoneID = ZoneId.of("America/Chicago");
ZonedDateTime zdt = ZonedDateTime.of(
    LocalDate.of(2021, 11, 7),
    LocalTime.of(1, 30),
    zoneID
);
ZonedDateTime anHourLater = zdt.plusHours(1);
System.out.println(zdt.getHour() == anHourLater.getHour());
System.out.print(zdt.getOffset().equals(anHourLater.getOffset()));
```

What is the output?

- A. true false
- B. False false
- C. true true
- D. false true

**Answer:** A

**Explanation:**

The answer is A because the code fragment uses the ZoneId and ZonedDateTime classes to create two date-time objects with the same local date-time but different zone offsets. The ZoneId class represents a time-zone ID, such as America/Chicago, and the ZonedDateTime class represents a date-time with a time-zone in the ISO-8601 calendar system. The code fragment creates two ZonedDateTime objects with the same local date-time of 2021-11-07T01:30, but different zone IDs of America/Chicago and UTC. The code fragment then compares the two objects using the equals and isEqual methods.

The equals method compares the state of two objects for equality. In this case, it compares the local date-time, zone offset, and zone ID of the two ZonedDateTime objects. Since the zone offsets and zone IDs are different, the equals method returns false.

The isEqual method compares the instant of two temporal objects for equality. In this case, it compares the instant of the two ZonedDateTime objects, which is derived from the local date-time and zone offset. Since DST in Chicago ended on November 7th at 2 AM in 2021, the local date-time of 2021-11-07T01:30 in America/Chicago corresponds to the same instant as 2021-11-07T06:30 in UTC. Therefore, the isEqual method returns true.

Hence, the output is true false. References:

- ? Oracle Certified Professional: Java SE 17 Developer
- ? Java SE 17 Developer
- ? OCP Oracle Certified Professional Java SE 17 Developer Study Guide
- ? ZoneId (Java Platform SE 8 )
- ? ZonedDateTime (Java Platform SE 8 )
- ? Time Zone & Clock Changes in Chicago, Illinois, USA
- ? Daylight Saving Time Changes 2023 in Chicago, USA

**NEW QUESTION 44**

Given the code fragment:

```
// Login time:2021-01-12T21:58:18.817Z
Instant loginTime = Instant.now();
Thread.sleep(1000);

// Logout time:2021-01-12T21:58:19.880Z
Instant logoutTime = Instant.now();

loginTime = loginTime.truncatedTo(ChronoUnit.MINUTES);    // line n1
logoutTime = logoutTime.truncatedTo(ChronoUnit.MINUTES);

if (logoutTime.isAfter(loginTime))
    System.out.println("Logged out at: " + logoutTime);
else
    System.out.println("Can't logout");
```

What is the result?

- A. Logged out at: 2021-0112T21:58:19.880z
- B. Logged out at: 2021-01-12T21:58:00z
- C. A compilation error occurs at Line n1.
- D. Can't logout

**Answer: B**

**Explanation:**

The code fragment is using the Java SE 17 API to get the current time and then truncating it to minutes. The result will be the current time truncated to minutes, which is why option B is correct. References:

? [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>

? [https://docs.oracle.com/javase/17/docs/api/java.base/java/time/Instant.html#truncatedTo\(java.time.temporal.TemporalUnit\)](https://docs.oracle.com/javase/17/docs/api/java.base/java/time/Instant.html#truncatedTo(java.time.temporal.TemporalUnit))

**NEW QUESTION 48**

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 50**

Given:

```
public class Test {
    public String attach1(List<String> data) {
        return data.parallelStream().reduce("w", (n,m) -> n+m, String::concat);
    }
    public String attach2(List<String> data) {
        return data.parallelStream().reduce((l, p)-> l+p).get();
    }

    public static void main(String[] args) {
        Test t = new Test();
        var list = List.of("Table", "Chair");
        String x= t.attach1(list);
        String y= t.attach2(list);
        System.out.print(x+ " "+y);
    }
}
```

What is the result?

- A. Tablechair Tablechair
- B. Wtablechair tableChair
- C. A RuntimeException is thrown
- D. wTableChair TableChair
- E. Compilation fails

**Answer:** E

**Explanation:**

The code fragment will fail to compile because the class name and the constructor name do not match. The class name is Furniture, but the constructor name is Wtable. This will cause a syntax error. The correct way to define a constructor is to use the same name as the class name. Therefore, the code fragment should change the constructor name to Furniture or change the class name to Wtable.

**NEW QUESTION 51**

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 54**

.....

## THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual 1z0-829 Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the 1z0-829 Product From:

<https://www.2passeasy.com/dumps/1z0-829/>

## Money Back Guarantee

### 1z0-829 Practice Exam Features:

- \* 1z0-829 Questions and Answers Updated Frequently
- \* 1z0-829 Practice Questions Verified by Expert Senior Certified Staff
- \* 1z0-829 Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- \* 1z0-829 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year