Oracle

Exam Questions 1z0-829

Java SE 17 Developer

NEW QUESTION 1
Given:

```java
public class App {
    public int x = 100;

    public static void main(String[] args) {
        int x = 1000;
        App t = new App();
        t.myMethod(x);
        System.out.println(x);
    }
    public void myMethod(int x) {
        x++;
        System.out.println(x);
        System.out.println(this.x);
    }
}
```

What is the result?
A.
1001
1001
1000
B.
101
101
1000
C.
100
100
1000
D.
1001
100
1000

A.

Answer: D

Explanation:
The code fragment is using the bitwise operators & (AND), | (OR), and ^ (XOR) to perform operations on the binary representations of the integer values. The & operator returns a 1 in each bit position where both operands have a 1, the | operator returns a 1 in each bit position where either operand has a 1, and the ^ operator returns a 1 in each bit position where only one operand has a 1. The binary representations of the integer values are as follows:
? 1000 = 1111101000
? 100 = 1100100
? 101 = 1100101
The code fragment performs the following operations:
? x = x ^ y; // x becomes 1111010101, which is 1001 in decimal
? y = x ^ y; // y becomes 1100100, which is 100 in decimal
? x = x ^ y; // x becomes 1100101, which is 101 in decimal
The code fragment then prints out the values of x, y, and z, which are 1001, 100, and 1000 respectively. Therefore, option D is correct.

NEW QUESTION 2
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 3
Given the code fragment:

```
String a = "Hello! Java";
System.out.print(a.indexOf("Java"));
a.replace("Hello!", "Welcome!");
System.out.print(a.indexOf("Java"));
StringBuilder b = new StringBuilder(a);
System.out.print(b.indexOf("Java"));
```

What is the result?

A. 81111
B. 8109
C. 777
D. 71010
E. 888
F. 7107

Answer: B

Explanation:
The code fragment is creating a string variable ??a?? with the value ??Hello! Java??. Then, it is printing the index of ??Java?? in ??a??. Next, it is replacing ??Hello!?? with ??Welcome!?? in ??a??. Then, it is printing the index of ??Java?? in ??a??. Finally, it is creating a new StringBuilder object ??b?? with the value of ??a?? and printing the index of ??Java?? in ??b??. The output will be 8109 because the index of ??Java?? in ??a?? is 8, the index of ??Java?? in ??a?? after replacing ??Hello!?? with ??Welcome!?? is 10, and the index of ??Java?? in ??b?? is 9. References: Oracle Java SE 17 Developer source and documents: [String (Java SE 17 & JDK 17)], [StringBuilder (Java SE 17 & JDK 17)]

NEW QUESTION 4
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 5
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 6
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 7
Given the code fragment:

```
Duration duration = Duration.ofMillis(5000);
System.out.print(duration);
duration = Duration.ofSeconds(60);
System.out.print(duration);
Period period = Period.ofDays(6);
System.out.print(period);
```

What is the result?

A. $SIM6D
B. PT5000PT60MP6D
C. PT5SPTIMP6D
D. 5000$60M6D

Answer: B

Explanation:
 The code fragment is creating a Duration object with a value of 5000 milliseconds, then printing it. Then, it is creating another Duration object with a value of 60 seconds, then printing it. Finally, it is creating a Period object with a value of 6 days, then printing it. The output will be ??PT5000PT60MP6D??. References: https://docs.oracle.com/javase/8/docs/api/java/time/Duration.html, https://docs.oracle.com/javase/8/docs/api/java/time/Period.html

NEW QUESTION 8
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,CarImp1 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 9
Given the course table:

COURSE_ID	COURSE_NAME	COURSE_FEE	COURSE_LEVEL
1021	Java Programmer	400.00	1
1022	Java Architect	600.00	2
1023	Java Master	600.00	2

Given the code fragment:

```
try (Connection con = DriverManager.getConnection(connectionString)) {
    Statement statement = con.createStatement(TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPDATABLE);
    String qry = "UPDATE course SET course_fee = ? where COURSE_LEVEL = ?";
    PreparedStatement prStmt = con.prepareStatement(qry, TYPE_SCROLL_INSENSITIVE);
    prStmt.setDouble(1,600.00);
    prStmt.setInt(2,2);
    System.out.println(prStmt.executeUpdate());
}
catch(SQLException sqlException) {
    System.out.println(sqlException);
}
```

A. 2
B. false
C. true
D. 1

Answer: C

Explanation:
 The code fragment will execute the update statement and set the course fee of the course with ID 1021 to 5000. The executeUpdate method returns an int value that indicates the number of rows affected by the SQL statement. In this case, only one row will be updated, so the result variable will be 1. The if statement will check if the result is greater than 0, which is true, and print ??Updated successfully??. Therefore, the output of the code fragment is true. References: https://education.oracle.com/products/trackp_OCPJSE17, https://mylearn.oracle.com/ou/learning-path/java-se-17-developer/99487, https://docs.oracle.com/en/java/javase/17/docs/api/java.sql/java/sql/Statement.html#execut eUpdate(java.lang.String)

NEW QUESTION 10
Given the content of the in. tart 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 10
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 initialized1. 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 11
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 12
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 13
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 17
......

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

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

100% Actual & Verified — Instant Download, Please Click
Order The 1z0-829 Practice Test Here