



**Vendor:** Oracle

**Exam Code:** 1Z0-805

**Exam Name:** Upgrade to Java SE 7 Programmer

**Version:** DEMO

### QUESTION 1

Given the following code fragment:

```
public class Calc {
    public static void main (String [] args) {
        /* insert code here Line **
        System.out.print("The decimal value is" + var);
    }
}
```

Which three code fragments, when inserted independently at line \*\*, enable the code to compile/

- A. `int var = 0b_1001;`
- B. `long var = 0b100_01L;`
- C. `float var = 0b10_01;`
- D. `float var = 0b10_01F;`
- E. `double var = 0b10_01;`
- F. `double var = 0b10_01D;`

**Answer:** BCE

**Explanation:**

B: output 17

C: output 9.0

E: output 9.0

Not A: A \_ character cannot begin a number.

Not D: A float cannot be defined as a binary number (with literal B) Not F: A float cannot be defined as a decimal number (with literal D)

Note1:

In Java SE 7 and later, any number of underscore characters ( \_ ) can appear anywhere between digits in a numerical literal. This feature enables you, for example, to separate groups of digits in numeric literals, which can improve the readability of your code.

For instance, if your code contains numbers with many digits, you can use an underscore character to separate digits in groups of three, similar to how you would use a punctuation mark like a comma, or a space, as a separator.

You can place underscores only between digits; you cannot place underscores in the following places:

- \* At the beginning or end of a number (not A)
- \* Adjacent to a decimal point in a floating point literal
- \* Prior to an F or L suffix
- \* In positions where a string of digits is expected

Note 2: An integer literal is of type long if it ends with the letter L or l; otherwise it is of type int. It is recommended that you use the upper case letter L because the lower case letter l is hard to distinguish from the digit 1.

Values of the integral types byte, short, int, and long can be created from int literals. Values of type long that exceed the range of int can be created from long literals. Integer literals can be expressed by these number systems:

Decimal: Base 10, whose digits consists of the numbers 0 through 9; this is the number system you use every day

Hexadecimal: Base 16, whose digits consist of the numbers 0 through 9 and the letters A through F

Binary: Base 2, whose digits consists of the numbers 0 and 1 (you can create binary literals in

Java SE 7 and later)

Reference: The Java Tutorials, Primitive Data Types:  
Using Underscore Characters in Numeric Literals

### QUESTION 2

Given the code fragment:

```
String query = "SELECT ID FROM Employee"; \\ Line 1
try (Statement stmt = conn.createStatement()) { \\ Line 2
    ResultSet rs = stmt.executeQuery(query); \\ Line 3
    stmt.executeQuery ("SELECT ID FROM Customer"); \\ Line 4
    while (rs.next()) {
        \\process the results
        System.out.println ("Employee ID: " + rs.getInt("ID" ));
    }
} catch (Exception e) {
    system.out.println ("Error");
}
```

Assume that the SQL queries return records. What is the result of compiling and executing this code fragment?

- A. The program prints employees IDs.
- B. The program prints customer IDs.
- C. The program prints Error.
- D. Compilation fails on line 13.

**Answer:** A

**Explanation:**

Line 3 sets the resultset rs. rs will contain IDs from the employee table. Line 4 does not affect the resultset rs. It just returns a resultset (which is not used).

Note:

A ResultSet object is a table of data representing a database result set, which is usually generated by executing a statement that queries the database.

You access the data in a ResultSet object through a cursor. Note that this cursor is not a database cursor. This cursor is a pointer that points to one row of data in the ResultSet. Initially, the cursor is positioned before the first row. The method ResultSet.next moves the cursor to the next row. This method returns false if the cursor is positioned after the last row. This method repeatedly calls the ResultSet.next method with a while loop to iterate through all the data in the ResultSet.

Reference: The Java Tutorials, Retrieving and Modifying Values from Result Sets

### QUESTION 3

Given:

```
public class SampleClass {
    public static void main(String[] args) {
        SampleClass sc = new SampleClass();
        sc.processCD();
    }
}
```

```
private void processCD() {
try (CDStream cd = new CDStream()) {
cd.open();
cd.read();
cd.write("lullaby");
cd.close();
} catch (Exception e) {
System.out.println("Exception thrown");
}
}
class CDStream {
String cdContents = null;
public void open() {
cdContents = "CD Contents";
System.out.println("Opened CD stream");
}
public String read() throws Exception {
throw new Exception("read error");
}
public void write(String str) {
System.out.println("CD str is: " + str);
}
public void close() {
cdContents = null;
}
}
```

What is the result?

- A. Compilation CD stream
- B. Opened CD thrown
- C. Exception thrown
- D. Opened CD stream
- E. CD str is: lullaby

**Answer:** A

**Explanation:**

In this example the compilation of line " try (CDStream cd = new CDStream()) {" will fail, as try-with-resources not applicable to variable type CDStream.

Note: The try-with-resources statement is a try statement that declares one or more resources. A resource is an object that must be closed after the program is finished with it. The try-with-resources statement ensures that each resource is closed at the end of the statement. Any object that implements `java.lang.AutoCloseable`, which includes all objects which implement `java.io.Closeable`, can be used as a resource.

Reference: The Java Tutorials, The try-with-resources Statement

#### QUESTION 4

Given the code fragment:

```
SimpleDateFormat sdf;
```

Which code fragment displays the three-character month abbreviation?

- A. `sdf = new SimpleDateFormat ("mm", Locale.UK);  
System.out.println ("Result:"+ sdf.format(new Date()));`
- B. `sdf = new SimpleDateFormat ("MM", Locale.UK);  
System.out.println ("Result:"+ sdf.format(new Date()));`
- C. `sdf = new SimpleDateFormat ("MMM", Locale.UK);  
System.out.println ("Result:"+ sdf.format(new Date()));`
- D. `sdf = new SimpleDateFormat ("MMMM", Locale.UK);  
System.out.println ("Result:"+ sdf.format(new Date()));`

**Answer: C**

**Explanation:**

C: Output example: Apr

Note:SimpleDateFormat is a concrete class for formatting and parsing dates in a locale-sensitive manner. It allows for formatting (date -> text), parsing (text -> date), and normalization.

SimpleDateFormat allows you to start by choosing any user-defined patterns for date-time formatting. However, you are encouraged to create a date-time formatter with either `getTimeInstance`, `getDateInstance`, or `getDateTimelInstance` in `DateFormat`. Each of these class methods can return a date/time formatter initialized with a default format pattern. You may modify the format pattern using the `applyPattern` methods as desired.

#### QUESTION 5

Given the code fragment:

```
public static void processFile () throws IOException {  
    Try (FileReader fr = new FileReader ("logfilesrc.txt");  
    FileWriter fw = new FileWriter ("logfilesdst.txt") ) {  
        int i = fr.read();  
    }  
}
```

Which statement is true?

- A. The code fragment contains compilation errors.
- B. The java runtime automatically closes the FileWriter Instance first and the FileReader instance next.
- C. The java runtime automatically closes the FileReader Instance first and the FileWriter instance next.
- D. The developer needs to close the FileReader instance first and the FileWriter instance explicitly in a catch block.
- E. The Java runtime may close the FileReader and FileWriter instance in an intermediate manner. Developers should not rely on the order in which they are closed.

**Answer: B**

**Explanation:**

The try-with-resources statement is a try statement that declares one or more resources. A resource is an object that must be closed after the program is finished with it. The try- with-resources statement ensures that each resource is closed at the end of the statement. Any object that implements `java.lang.AutoCloseable`, which includes all objects which implement `java.io.Closeable`, can be used as a resource.

Reference: The Java Tutorials,The try-with-resources Statement

### QUESTION 6

Given this code fragment:

```
ResultSet rs = null;
try (Connection conn = DriverManager. getConnection (url) ) {
Statement stmt = conn.createStatement();
rs stmt.executeQuery(query);
//.. other methods }
} catch (SQLException se) {
System.out.println ("Error");
}
```

Which object is valid after the try block runs?

- A. The Connection object only
- B. The Statement object only
- C. The Result set object only
- D. The Statement and Result Set object only
- E. The connection, statement, and ResultSet objects
- F. Neither the Connection, Statement, nor ResultSet objects

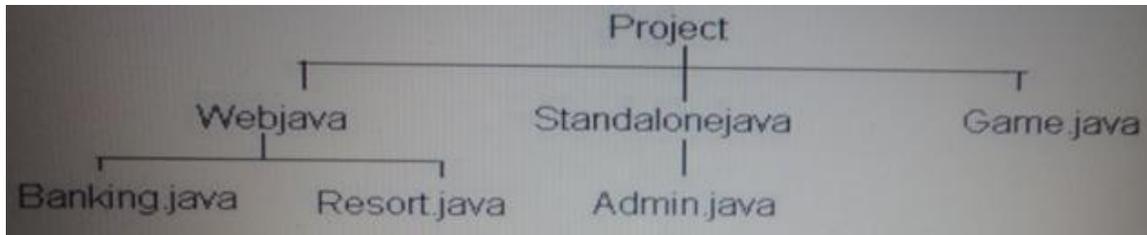
**Answer: C**

#### Explanation:

Generally, JavaScript has just 2 levels of scope: global and function. But, try/catch is an exception (no punn intended). When an exception is thrown and the exception object gets a variable assigned to it, that object variable is only available within the "catch" section and is destroyed as soon as the catch completes.

### QUESTION 7

View the Exhibit:



Given the following code fragment:

```
class Finder extends SimpleFileVisitor<Path> {
private final PathMatcher matcher;
private static int numMatches = 0;
Finder() {
matcher = FileSystems.getDefault().getPathMatcher("glob:*java");
}
void find(Path file) {
Path Name = file.getFileName();
if (name != null && matcher.matches(name)) {
```

```
numMatches++;
}
}
void report()
{
System.out.println("Matched: " + numMatches);
}
@Override
public FileVisitResult visitFile(Path file, BasicFileAttributes attrs) {
find(file);
return CONTINUE;
}
}
public class Visitor {
public static void main(String[] args) throws IOException {
Finder finder = new Finder();
Files.walkFileTree(Paths.get("d:\\Project"), finder);
finder.report();
}
}
```

What is the result?

- A. Compilation fails
- B. 6
- C. 4
- D. 1
- E. 3

**Answer: B**

**Explanation:**

The program will compile and run.

Referring to the exhibit there will be six nodes that matches glob:\*java.

## Thank You for Trying Our Product

### Braindump2go Certification Exam Features:

- ★ More than **99,900** Satisfied Customers Worldwide.
- ★ Average **99.9%** Success Rate.
- ★ **Free Update** to match latest and real exam scenarios.
- ★ **Instant Download** Access! No Setup required.
- ★ Questions & Answers are downloadable in **PDF** format and **VCE** test engine format.
- ★ Multi-Platform capabilities - **Windows, Laptop, Mac, Android, iPhone, iPod, iPad.**
- ★ **100%** Guaranteed Success or **100%** Money Back Guarantee.
- ★ **Fast**, helpful support **24x7.**



View list of all certification exams: <http://www.braindump2go.com/all-products.html>



Microsoft



ORACLE



CITRIX



JUNIPER  
NETWORKS



EMC<sup>2</sup>  
where information lives<sup>®</sup>

**10% Discount Coupon Code: BDNT2014**