

# 1Z0-816<sup>Q&As</sup>

Java SE 11 Programmer II

## Pass Oracle 1Z0-816 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass2lead.com/1z0-816.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by Oracle  
Official Exam Center

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers



### QUESTION 1

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

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

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

Correct Answer: B

### QUESTION 2

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

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

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

6104 3

Which is a proper JDBC URL?

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

Correct Answer: D

Reference: <https://vladmihalcea.com/jdbc-driver-connection-url-strings/>

---

### QUESTION 3

Which code fragment does a service use to load the service provider with a Print interface?

- A. `private Print print = com.service.Provider.getInstance();`
- B. `private java.util.ServiceLoader loader = ServiceLoader.load(Print.class);`
- C. `private java.util.ServiceLoader loader = new java.util.ServiceLoader();`
- D. `private Print print = new com.service.Provider.PrintImpl();`

Correct Answer: B

Reference: <https://docs.oracle.com/javase/8/docs/api/?java/util/ServiceLoader.html>

---

### QUESTION 4

Given:

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

What is the result?

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

Correct Answer: C

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

**Result**

compiled and executed in 1.227 sec(s)

```
Orange Juice
Apple Pie
Lemmon Ice
raspberry Tart
```

**QUESTION 5**

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

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

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

D. Stream lines = Files.lines(INPUT\_FILE\_NAME);

Correct Answer: C

[1Z0-816 PDF Dumps](#)

[1Z0-816 Study Guide](#)

[1Z0-816 Exam Questions](#)