



# 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.pass4itsure.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**

Examine these module declarations:

```
module ServiceAPI {
    exports com.example.api;
}

module ServiceProvider {
    requires ServiceAPI;
    provides com.example.api with com.myimpl.Impl;
}

module Consumer {
    requires ServiceAPI;
    uses com.example.api;
}
```

Which two statements are correct? (Choose two.)

- A. The ServiceProvider module is the only module that, at run time, can provide the com.example.api API.
- B. The placement of the com.example.api API in a separate module, ServiceAPI, makes it easy to install multiple provider modules.
- C. The Consumer module should require the ServiceProvider module.
- D. The ServiceProvider module should export the com.myimpl package.
- E. The ServiceProvider module does not know the identity of a module (such as Consumer) that uses the com.example.api API.

Correct Answer: AC

---

**QUESTION 2**

Given:



```
public interface TestInterface {
    default void samplingProbeProcedure() {
        probeProcedure();
        System.out.println("Collect Sample");
        System.out.println("Leave Asteroid");
        System.out.println("Dock with Main Craft");
    }
    default void explosionProbeProcedure() {
        probeProcedure();
        System.out.println("Explode")
    }
}
```

Examine these requirements:

Eliminate code duplication.

Keep constant the number of methods other classes may implement from this interface.

Which method can be added to meet these requirements?

- A. 

```
private default void probeProcedure() {
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}
```
- B. 

```
static void probeProcedure() {
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}
```
- C. 

```
private void probeProcedure() {
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}
```
- D. 

```
default void probeProcedure() {
    system.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}
```

A. Option A

B. Option B

C. Option C

D. Option D



Correct Answer: B

---

### QUESTION 3

Given: What is the result?

```
int arr[][] = {{5,10},{8,12},{9,3}};
long count = Stream.of(arr)
    .flatMapToInt(IntStream::of)
    .map(n -> n + 1)
    .filter(n -> (n % 2 == 0))
    .peek(System.out::print)
    .count();
System.out.println(" " + count);
```

- A. 6910 3
- B. 10126 3
- C. 3
- D. 6104 3

Correct Answer: D

---

### QUESTION 4

Given:

```
// line 1
List<String> fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
fruits.replaceAll(function);
```

Which statement on line 1 enables this code fragment to compile?

- A. Function function = String::toUpperCase;
- B. UnaryOperator function = s -> s.toUpperCase();
- C. UnaryOperator function = String::toUpperCase;
- D. Function function = m -> m.toUpperCase();

Correct Answer: C

---



```
1
2 import java.io.*;
3 import java.util.*;
4 import java.util.stream.Stream;
5 import java.util.function.Function;
6 import java.util.function.UnaryOperator;
7
8 class Hello {
9     public static void main(String[] args) {
10
11         UnaryOperator<String> function = String::toUpperCase;
12         List<String> fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
13         fruits.replaceAll(function);
14
15     }
16 }
17
```

### QUESTION 5

Given:

```
class CustomType<T> {
    public <T> int count(T[] anArray, T element) {
        int count = 0;
        for(T e : anArray) {
            if (e.equals(element)) ++count;
        }
        return count;
    }
}
```

and

```
public class Test extends CustomType {
    public static void main(String[] args) {
        String[] words = {"banana", "orange", "apple", "lemon"};
        Integer[] numbers = {1, 2, 3, 4, 5};
        CustomType type = new CustomType();
        CustomType<String> stringType = new CustomType<>();
        System.out.println(stringType.count(words, "apple"));
        System.out.println(type.count(words, "apple"));
        System.out.println(type.count (numbers, 3));
    }
}
```

What is the result?

- A. A NullPointerException is thrown at run time.
- B. The compilation fails.



C. 1 Null null

D. 1

E. A ClassCastException is thrown at run time.

Correct Answer: B

Console 4

```
Error: Could not find or load main class CustomType  
Caused by: java.lang.ClassNotFoundException: CustomType
```

[Latest 1Z0-816 Dumps](#)

[1Z0-816 Study Guide](#)

[1Z0-816 Braindumps](#)