



1Z0-819^{Q&As}

Java SE 11 Developer

Pass Oracle 1Z0-819 Exam with 100% Guarantee

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

<https://www.pass4itsure.com/1z0-819.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**

Which code fragment prints 100 random numbers?

- A. `var r= new Random();
new DoubleStream(r::nextDouble).limit(100).forEach(System.out::print);`
- B. `DoubleStream.generate(Random::nextDouble)
.limit(100).forEach(System.out::print);`
- C. `Doublestream.generate(Random.nextDouble).limit(100).forEach(System.out.print);`
- D. `var r = new Random(); DoubleStream.generate(r::nextDouble).limit(100).forEach(System.out::print);`

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: D

Reference: <https://www.javacodegeeks.com/2014/01/java-util-random-in-java-8.html>

QUESTION 2

Given:

```
public class FunctionalInterfaceTest {  
    public static void main(String[] args) {  
        List fruits = Arrays.asList("apple", "orange", "banana");  
        Consumer<String> c = System.out::print;  
        Consumer<String> output = c.andThen(x -> System.out.println(":" + x.toUpperCase  
    ));  
        fruits.forEach(output);  
    }  
}
```

What is the output?

- A. :APPLE:ORANGE:BANANA appleorangebanana
- B. :APPLE:ORANGE:BANANA
- C. APPLE:apple ORANGE:orange BANANA:banana
- D. appleorangebanana :APPLE:ORANGE:BANANA
- E. apple:APPLE orange:ORANGE banana:BANANA

Correct Answer: E



QUESTION 3

Given:

```
public class Main {
    public static void main(String[] args) {
        var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
        Optional<Integer> result = numbers.stream().filter(x -> x % 3 != 0).reduce((i, j)
-> i + j);
        result.ifPresent(System.out::print); // line 1
    }
}
```

Which is true about line 1?

- A. If the value is not present, a NoSuchElementException is thrown at run time.
- B. It always executes the System.out::print statement.
- C. If the value is not present, a NullPointerException is thrown at run time.
- D. If the value is not present, nothing is done.

Correct Answer: D

```
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 import java.util.Optional;
11
12
13 public class Main {
14     public static void main(String[] args) {
15         var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
16         Optional<Integer> result = numbers.stream().filter(x -> x % 3 != 0).reduce((i, j) -> i + j);
17
18     }
19 }
```

Result

CPU Time: 0.18 sec(s), Memory: 33380 kilobyte(s)

```
]Doodle in Action... Running the program...
```

QUESTION 4

Which two commands are used to identify class and module dependencies? (Choose two.)

- A. java --show-module-resolution



- B. jmod describe
- C. java Hello.java
- D. jdeps --list-deps
- E. jar --show-module-resolution

Correct Answer: BD

The two commands that can be used to identify class and module dependencies are B. jmod describe and D. jdeps --list-deps. jmod describe is a command that can be used to display information about the contents of a jmod file, including its dependencies. jdeps --list-deps is a command that can be used to list the dependencies of a Java class file. jdeps is a command-line tool that can be used to analyze the static dependencies of Java applications and libraries.

QUESTION 5

```
1 public class Test {
2     private static class Greet {
3         private void print() {
4             System.out.println("Hello World");
5         }
6     }
7     public static void main(String[] args) {
8         Test.Greet i = new Greet();
9         i.print();
10    }
11 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32504 kilobyte(s)

Hello World

Which two describe reasons to modularize the JDK? (Choose two.)

- A. easier to understand the Java language



- B. improves security and maintainability
- C. easier to expose implementation details
- D. improves application robustness
- E. easier to build a custom runtime linking application modules and JDK modules

Correct Answer: BD

QUESTION 6

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 7

Given:

```
import java.util.ArrayList;
import java.util.Arrays;
public class NewMain {
    public static void main(String[] args) {
        String[] catNames = { "abyssinian", "oxicat",
            "korat", "laperm", "bengal", "sphynx" };
        var cats = new ArrayList<>(Arrays.asList(catNames));
        cats.sort((var a, var b) -> -a.compareTo(b));
        cats.forEach(System.out::println);
    }
}
```

What is the result?

A. abyssinian oxicat korat laperm bengal sphynx



- B. abyssinian bengal korat laperm oxicat sphynx
- C. sphynx oxicat laperm korat bengal abyssinian
- D. nothing

Correct Answer: C

QUESTION 8

Given:

```
StringBuilder s = new StringBuilder("ABCD");
```

Which would cause s to be AQCD?

- A. `s.replace(s.indexOf("A"), s.indexOf("C"), "Q");`
- B. `s.replace(s.indexOf("B"), s.indexOf("C"), "Q");`
- C. `s.replace(s.indexOf("B"), s.indexOf("B"), "Q");`
- D. `s.replace(s.indexOf("A"), s.indexOf("B"), "Q");`

Correct Answer: B

QUESTION 9

Given the code fragment:



```
public class Test{
    public static void main(String[] args) {
        try {
            if ("oracle".equals("ORACLE".toLowerCase())) {
                throw new NoMatchException();
            }
        } catch (NoMatchException | NullPointerException npe) {
            System.out.println("Exception 1");
        } catch (RuntimeException e) {
            System.out.println("Exception 2");
        } catch (Exception e) {
            System.out.println("Exception 3");
        } finally {
            System.out.println("Finally Block");
        }
    }
}
```

How many lines of text does this program print?

- A. four
- B. one
- C. three
- D. two

Correct Answer: D

QUESTION 10

Given:

```
public class Employee {
    private String name;
    private String locality;
    /* the constructor, getter and setter methods code goes here */
}
```

and:



```
8. List<Employee> roster = new ArrayList<>();
9. long empCount = roster.stream()
10. /* insert code here */
11. System.out.print(empCount);
```

Which code, when inserted on line 10, prints the number of unique localities from the roster list?

- A. `.map(Employee::getLocality) .distinct() .count();`
- B. `map(e > e.getLocality()) .count();`
- C. `.map(e > e.getLocality()) .collect(Collectors.toSet()) .count();`
- D. `.filter(Employee::getLocality) .distinct() .count();`

Correct Answer: D

Reference: <https://developer.android.com/reference/android/location/Address>

QUESTION 11

Given:

```
public class Employee {
    private String name;
    private String neighborhood;
    private LocalDate birthday;
    private int salary;
}

and

List<Employee> roster = new ArrayList<>(...);
Map<String, Optional<Employee>> m = roster.stream()
// Line 1
```

Which code fragment on line 1 makes the m map contain the employee with the highest salary for each neighborhood?

- A. `.collect(Collectors.maxBy(Employee::getSalary, Collectors.groupingBy(Comparator.comparing(e -> e.getNeighborhood()))));`
- B. `.collect(Collectors.groupingBy(Employee::getNeighborhood, Collectors.maxBy(Comparator.comparing(Employee::getSalary))));`
- C. `.collect(Collectors.groupingBy(e -> e.getNeighborhood(), Collectors.maxBy((x, y) -> y.getSalary() - x.getSalary())));`
- D. `.collect(Collectors.maxBy((x, y) -> y.getSalary() - x.getSalary(), Collectors.groupingBy(Employee::getNeighborhood)));`

A. Option A



B. Option B

C. Option C

D. Option D

Correct Answer: C

QUESTION 12

Given:

```
for(var i = 0; i < 10; i++) {  
    switch(i%5) {  
        case 2:  
            i *= i;  
            break;  
        case 3:  
            i++;  
            break;  
        case 1:  
        case 4:  
            i++;  
            continue;  
        default:  
            break;  
    }  
    System.out.print(i + " ");  
    i++;  
}
```

What is the result?

A. The code prints nothing.

B. 0

C. 10

D. 0 4 9

E. 0 8

Correct Answer: E

QUESTION 13

Given the code fragment:



```
char[][] arrays = {{'a', 'd'}, {'b', 'e'}, {'c', 'f'}};
for (char[] xx : arrays) {
    for (char yy : xx) {
        System.out.print(yy);
    }
    System.out.print(" ");
}
```

What is the result?

- A. ab cd ef
- B. An `ArrayIndexOutOfBoundsException` is thrown at runtime.
- C. The compilation fails.
- D. abc def
- E. ad be cf

Correct Answer: E

QUESTION 14

Given:

```
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);
```

What is the result?

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

Correct Answer: D



QUESTION 15

Which two statements correctly describe capabilities of interfaces and abstract classes? (Choose two.)

- A. Interfaces cannot have protected methods but abstract classes can.
- B. Both interfaces and abstract classes can have final methods.
- C. Interfaces cannot have instance fields but abstract classes can.
- D. Interfaces cannot have static methods but abstract classes can.
- E. Interfaces cannot have methods with bodies but abstract classes can.

Correct Answer: AC

Reference: <https://www.guru99.com/interface-vs-abstract-class-java.html>

[1Z0-819 Study Guide](#)

[1Z0-819 Exam Questions](#)

[1Z0-819 Braindumps](#)