

DUMPSBOSS.

Java SE 11 Developer

Oracle 1z0-819

Version Demo

Total Demo Questions: 15

Total Premium Questions: 248

Buy Premium PDF

<https://dumpsboss.co>

support@dumpsboss.co

support@dumpsboss.co

dumpsboss.co

QUESTION NO: 1

Given:

```
int i = 0;
for( ; i<10; i++){
    System.out.print(++i + " ");
}
```

Which two statements are valid to be written in this interface? (Choose two.)

- A. public abstract void methodB();
- B. final void methodG(){System.out.println("G");}
- C. private abstract void methodC();
- D. public String methodD();
- E. public int x;
- F. final void methodE();
- G. public void methodF(){System.out.println("F");}

ANSWER: A D

QUESTION NO: 2

Given the code fragment:

```
String s1 = new String("ORACLE");
String s2 = "ORACLE";
String s3 = s1.intern();

System.out.print((s1==s2) + " ");
System.out.print((s2==s3) + " ");
System.out.println(s1==s3);
```

What is the result?

- A. false true true
- B. true false false

C. false false true

D. false true false

ANSWER: D

QUESTION NO: 3

Given an application with a main module that has this module-info.java file:

```
module main {  
    exports country;  
    uses country.CountryDetails;  
}
```

Which two are true? (Choose two.)

- A. A module providing an implementation of `country.CountryDetails` can be compiled and added without recompiling the main module.
- B. A module providing an implementation of `country.CountryDetails` must have a `requires main;` directive in its `module-info.java` file.
- C. An implementation of `country.countryDetails` can be added to the main module.
- D. To compile without an error, the application must have at least one module in the module source path that provides an implementation of `country.CountryDetails`.
- E. To run without an error, the application must have at least one module in the module path that provides an implementation of `country.CountryDetails`.

ANSWER: B D

QUESTION NO: 4

Given:

```
import java.util.*;

public class Main {
    static Map<String, String> map = new HashMap<>();
    static List<String> keys =
        new ArrayList<>(List.of("S", "P", "Q", "R"));
    static String[] values =
        {"senate", "people", "of", "rome" };

    static {
        for(var i = 0; i < keys.size(); i++) {
            map.put(keys.get(i), values[i]);
        }
    }

    public static void main(String[] args) {
        keys.clear();
        values = new String[0];
        System.out.println("Keys: " + keys.size() +
            " Values: " + values.length +
            " Map: " + map.size());
    }
}
```

What is the result?

- A. Keys: 4 Values: 4 Map: 0
- B. Keys: 4 Values: 4 Map: 4
- C. The compilation fails.
- D. Keys: 0 Values: 0 Map:
- E. Keys: 0 Values: 0 Map: 0

ANSWER: B

QUESTION NO: 5

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

ANSWER: E

QUESTION NO: 6

Given:

```
List<String> list1 = new LinkedList<String>();
Set<String> hs1 = new HashSet<String>();
String[] v = {"a", "b", "c", "b", "a"};
for (String s: v) {
    list1.add(s);
    hs1.add(s);
}
System.out.print(hs1.size() + " " + list1.size() + " ");
HashSet hs2 = new HashSet(list1);
LinkedList list2 = new LinkedList(hs1);
System.out.print(hs2.size() + " " + list2.size());
```

What is the result?

- A. 3 5 3 3
- B. 3 3 3 3
- C. 3 5 3 5
- D. 5 5 3 3

ANSWER: A

QUESTION NO: 7

Given the code fragment:

```
9. Integer[] ints = {1,2,3,4,5,6,7};
10. var list = Arrays.asList(ints);
11. UnaryOperator<Integer> uo = x -> x * 3;
12. list.replaceAll(uo);
```

Which can replace line 11?

- A. UnaryOperator uo = (var x) -> (x * 3);
- B. UnaryOperator uo = var x -> { return x * 3; };
- C. UnaryOperator uo = x -> { return x * 3; };
- D. UnaryOperator uo = (int x) -> x * 3;

ANSWER: A

QUESTION NO: 8

Given the code fragment:

```
8. public class Test {
9.     private final int x = 1;
10.    static final int y;
11.    public Test() {
12.        System.out.print(x);
13.        System.out.print(y);
14.    }
15.    public static void main(String args[]) {
16.        new Test();
17.    }
18. }
```

What is the result?

- A. 1
- B. The compilation fails at line
- C. 10
- D. The compilation fails at line 16.
- E. The compilation fails at line 13.

ANSWER: C

QUESTION NO: 9

Which three initialization statements are correct? (Choose three.)

- A. `int x = 12_34;`
- B. `short sh = (short)'A';`
- C. `String contact# = "(+2) (999) (232)";`
- D. `boolean true = (4 == 4);`
- E. `float x = 1.99;`
- F. `int[][] e = {{1,1},{2,2}};`
- G. `byte b = 10;char c = b;`

ANSWER: A B F

QUESTION NO: 10

Given:

```
public interface Interfaceone (  
void printone(),
```

Which three classes successfully override printOne()? (Choose three.)

```
public abstract class TestClass implements Interfaceone ( public abscracc void printone()
```

B.

```
public class TestClass implements InterfaceOne (  
private void printone(){
```

```
System.out.println("one");
```

```
public class TestClass implements InterfaceOr.e (
```

```
public void printone(){
```

```
System.out.println("one");
```

D.

```
n.public abstract class TestClass implements Interfaceone (
```

```
nprintone(){
```

```
System.out.println('one');
```

E.

```
public abstract class TestClass implements InterfaceOne {  
    public String printOne() {  
        return "one";  
    }  
}
```

F.

```
public class TestClass{  
    public void printOne(){  
        System.out.println("one");  
    }  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

ANSWER: A C D

QUESTION NO: 11

Given:

```
public class Test {
    public static void main(String[] args) {
        int x;
        int y = 5;
        if (y > 2) {
            x = ++y;
            y = x + 7;
        } else {
            y++;
        }
        System.out.print(x + " " + y);
    }
}
```

What is the result?

- A. compilation error
- B. 0 5
- C. 6 13
- D. 5 12

ANSWER: A

QUESTION NO: 12

Given:

```
public interface A {
    public Iterable a();
}
public interface B extends A {
    public Collection a();
}
public interface C extends A {
    public Path a();
}
public interface D extends B, C {
}
```

Why does D cause a compilation error?

- A. D inherits a() only from C.
- B. D inherits a() from B and C but the return types are incompatible.
- C. D extends more than one interface.
- D. D does not define any method.

ANSWER: B

QUESTION NO: 13

Analyze the code:

```
public class Test {  
    static String prefix = "Global:";  
    private String name = "namespace";  
    public static String getName() {  
        return new Test().name;  
    }  
    public static void main(String[] args) {  
        Test t = new Test();  
        System.out.println(/* Insert code here */);  
    }  
}
```

Which two options can you insert inside println method to produce Global:namespace? (Choose two.)

- A. Test.prefix+Test.name
- B. new Test().prefix+new Test().name
- C. Test.prefix+Test.getName()
- D. Test.getName+prefix
- E. prefix+Test.name
- F. prefix+name

ANSWER: B C

QUESTION NO: 14

Given:

```
public class DNASynth {
    int aCount;
    int tCount;
    int cCount;
    int gCount;

    void setACount(int cCount){
        cCount = cCount;
    }
    void setTCount(){
        this.tCount = tCount;
    }
    int setCCount(){
        return cCount;
    }
    int setGCount(int g){
        gCount = g;
        return gCount;
    }
    void setAllCounts(int x){
        aCount = tCount = this.cCount = setGCount(x);
    }
}
```

Which two methods modify field values? (Choose two.)

- A. setAllCounts
- B. setACount
- C. setGCount
- D. setCCount
- E. setTCount

ANSWER: A C

QUESTION NO: 15

Assuming the Widget class has a getPrice method, this code does not compile:

```
List widgets = List.of(new Widget("Basic Widget", 19.55), // line 1
    new Widget("Enhanced Widget", 35.00),
    new Widget("Luxury Edition Widget", 55.45));
Stream widgetStream = widgets.stream(); // line 4
widgetStream.filter(a -> a.getPrice() > 20.00) // line 5
    .forEach(System.out::println);
```

Which two statements, independently, would allow this code to compile? (Choose two.)

- A. Replace line 5 with `widgetStream.filter(a > ((Widget)a).getPrice() > 20.00)`.
- B. Replace line 1 with `List widgetStream = widgets.stream();`.
- C. Replace line 5 with `widgetStream.filter((Widget a) > a.getPrice() > 20.00)`.
- D. Replace line 4 with `Stream widgetStream = widgets.stream();`.

ANSWER: A D