



1Z0-816^{Q&As}

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.geekcert.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

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();`.

Correct Answer: AD

QUESTION 2

Given: Which statement on line 1 enables this code to compile?

```
var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
// line 1
StringBuilder sb = new StringBuilder();
for(int a: numbers) {
    sb.append(f.apply(a));
    sb.append(" ");
}
System.out.println(sb.toString());
```

- A. Function `f = n -> n * 2;`
- B. Function `f = n -> n * 2;`
- C. Function `f = n -> n * 2;`
- D. Function `f = n -> n * 2;`
- E. Function `f = n -> n * 2;`

Correct Answer: A



```
15
16 public class Main {
17     public static void main(String[] args) {
18         var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
19         Function<Integer, Integer> f = n -> n * 2;
20         StringBuilder sb = new StringBuilder();
21         for(int a: numbers) {
22             sb.append(f.apply(a));
23             sb.append(" ");
24         }
25         System.out.println(sb.toString());
26     }
27 }
28
```

Result**CPU Time: 0.22 sec(s), Memory: 33056 kilobyte(s)****2 4 6 8 10 12 14 16 18 20****QUESTION 3**

A company has an existing sales application using a Java 8 jar file containing packages:

com.company.customer; com.company.customer.orders; com.company.customer.info; com.company.sales;
com.company.sales.leads; com.company.sales.closed; com.company.orders; com.company.orders.pending;
com.company.orders.shipped.

To modularize this jar file into three modules, customer, sales, and orders, which module-info.java would be correct?



- A.

```
module com.company.customer {
    opens com.company.customer;
}
module com.company.sales{
    opens com.company.sales;
}
module com.company.orders {
    opens com.company.orders;
}
```
- B.

```
module com.company.customer {
    exports com.company.customer;
}
module com.company.sales{
    exports com.company.sales;
}
module com.company.orders{
    exports com.company.orders;
}
```
- C.

```
module com.company.customer {
    requires com.company.customer;
}
module com.company.sales{
    requires com.company.sales;
}
module com.company.orders {
    requires com.company.orders;
}
```
- D.

```
module com.company.customer {
    provides com.company.customer;
}
module com.company.sales{
    provides com.company.sales;
}
module com.company.orders {
    provides com.company.orders;
}
```

- A. Option A
B. Option B
C. Option C
D. Option D



Correct Answer: C

Reference: <https://developer.ibm.com/tutorials/java-modularity-3/>

QUESTION 4

Consider this method declaration:

```
void setSessionUser(Connection conn, String user) throws SQLException {  
    Statement stmt = conn.createStatement();  
    String sql = <EXPRESSION>;  
    stmt.execute();  
}
```

- A) "SET SESSION AUTHORIZATION " + user
- B) "SET SESSION AUTHORIZATION " + stmt.enquoteIdentifier(user)

Is A or B the correct replacement for and why?

- A. A, because it sends exactly the value of user provided by the calling code.
- B. B, because enquoting values provided by the calling code prevents SQL injection.
- C. A and B are functionally equivalent.
- D. A, because it is unnecessary to enclose identifiers in quotes.
- E. B, because all values provided by the calling code should be enquoted.

Correct Answer: A

Reference:

[https://www.google.com/url?](https://www.google.com/url?sa=t&andrc=j&andq=andescr=s&andsource=web&andcd=4&andved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxAB&andurl=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%2Fletter%2Fen_US%2Fdb2s2e90.pdf&andusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec)

[sa=t&andrc=j&andq=andescr=s&andsource=web&andcd=4&andved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxAB&andurl=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%2Fletter%2Fen_US%2Fdb2s2e90.pdf&andusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec](https://www.google.com/url?sa=t&andrc=j&andq=andescr=s&andsource=web&andcd=4&andved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxAB&andurl=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%2Fletter%2Fen_US%2Fdb2s2e90.pdf&andusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec)

QUESTION 5

Given:

```
var fruits = List.of("apple", "orange", "banana", "lemon");
```

You want to examine the first element that contains the character n.

Which statement will accomplish this?



- A. String result = fruits.stream().filter(f -> f.contains("n")).findAny();
- B. fruits.stream().filter(f -> f.contains("n")).forEachOrdered(System.out::print);
- C. Optional result = fruits.stream().filter(f -> f.contains("n")).findFirst ();
- D. Optional result = fruits.stream().anyMatch(f -> f.contains("n"));

Correct Answer: B

```
1 import java.io.*;
2 import java.util.*;
3 public class abc {
4     public static void main(String[] args) {
5
6         var fruits = List.of("apple", "orange", "banana", "lemon");
7
8         fruits.stream().filter(f -> f.contains("n")).forEachOrdered(System.out::print);
9
10    }
11 }
12
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4 Interactive Stdin Input

CommandLine Arguments

Result

CPU Time: 0.19 sec(s), Memory: 33200 kilobyte(s)

```
orangebanana\lemon
```

[Latest 1Z0-816 Dumps](#)

[1Z0-816 Exam Questions](#)

[1Z0-816 Braindumps](#)