

# 1Z0-815<sup>Q&As</sup>

Java SE 11 Programmer I

**Pass Oracle 1Z0-815 Exam with 100% Guarantee**

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

<https://www.leads4pass.com/1z0-815.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:

```
import java.io.FileNotFoundException;
import java.io.IOException;

public class Tester {
    public static void main(String[] args) {
        try {
            doA();
        } //line 1
    }
    private static void doA() throws IOException, IndexOutOfBoundsException {
        if (false) {
            throw new FileNotFoundException();
        } else {
            throw new IndexOutOfBoundsException();
        }
    }
}
```

What must be added in line 1 to compile this class?

- A. `catch(IOException e) {}`
- B. `catch(FileNotFoundException | IndexOutOfBoundsException e) {}`
- C. `catch(FileNotFoundException | IOException e) {}`
- D. `catch(IndexOutOfBoundsException e) {} catch(FileNotFoundException e) {}`
- E. `catch(FileNotFoundException e) {} catch(IndexOutOfBoundsException e) {}`

Correct Answer: C

---

**QUESTION 2**

Given: `/code/a/Test.java` containing:

```
package a;
import b.Best;
public class Test {
    public static void main(String[] args) {
        Best b = new Best();
    }
}
```

and

`/code/b/Best.java`

containing:

package b;

```
public class Best { }
```

Which is the valid way to generate bytecode for all classes?

A. `java /code/a/Test.java`

B. `javac -d /code /code/a/Test`

C. `java /code/a/Test.java /code/b/Best.java`

D. `java -cp /code a.Test`

E. `javac -d /code /code/a/Test.java /code/b/Best.java`

F. `javac -d /code /code/a/Test.java`

Correct Answer: E

---

### QUESTION 3

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.

Correct Answer: D

---

## QUESTION 4

Given:

```
public interface EulerInterface {
    double getEulerValue();
}

public class EulerLambda {
    public static void main(String[] args) {
        EulerInterface myEulerInterface;
        myEulerInterface = () -> "2.71828";
        System.out.println("Value of Euler = " + myEulerInterface.getEulerValue());
    }
}
```

What is the result?

- A. It throws a runtime exception.
- B. Value of Euler = 2.71828
- C. The code does not compile.
- D. Value of Euler = "2.71828"

Correct Answer: C

---

## QUESTION 5

Given: Which statement is true?

```
package test;
import java.time.*;
public class Diary {
    private LocalDate now = LocalDate.now();
    public LocalDate getDate() {
        return now;
    }
}
```

and

```
package test;
public class Tester {
    public static void main(String[] args) {
        Diary d = new Diary();
        System.out.println(d.getDate());
    }
}
```

- A. Class Tester does not need to import java.time.LocalDate because it is already visible to members of the package test.
- B. All classes from the package java.time. are loaded for the class Diary.
- C. Only LocalDate class from java.time package is loaded.
- D. Tester must import java.time.LocalDate in order to compile.

Correct Answer: A

[1Z0-815 Practice Test](#)

[1Z0-815 Study Guide](#)

[1Z0-815 Braindumps](#)