

# 1Z0-809<sup>Q&As</sup>

Java SE 8 Programmer II

**Pass Oracle 1Z0-809 Exam with 100% Guarantee**

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

<https://www.leads4pass.com/1z0-809.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 the definition of the Country class:

```
public class country {  
  
    public enum Continent {ASIA, EUROPE}  
  
    String name;  
  
    Continent region;  
  
    public Country (String na, Continent reg) {  
  
        name = na, region = reg;  
  
    }  
  
    public String getName () {return name;}  
  
    public Continent getRegion () {return region;}  
  
}
```

and the code fragment:

```
List couList = Arrays.asList (  
  
    new Country ("Japan", Country.Continent.ASIA),  
  
    new Country ("Italy", Country.Continent.EUROPE),  
  
    new Country ("Germany", Country.Continent.EUROPE));  
  
Map regionNames = couList.stream ()  
  
    .collect(Collectors.groupingBy (Country ::getRegion,  
  
    Collectors.mapping(Country::getName, Collectors.toList())));  
  
System.out.println(regionNames);
```

- A. {EUROPE = [Italy, Germany], ASIA = [Japan]}
- B. {ASIA = [Japan], EUROPE = [Italy, Germany]}
- C. {EUROPE = [Germany, Italy], ASIA = [Japan]}
- D. {EUROPE = [Germany], EUROPE = [Italy], ASIA = [Japan]}

Correct Answer: B

---

**QUESTION 2**

Given:

```
IntStream stream = IntStream.of (1,2,3);
```

```
IntFunction inFu= x -> y -> x*y; //line n1
```

```
IntStream newStream = stream.map(inFu.apply(10)); //line n2
```

```
newStream.forEach(System.out::print);
```

Which modification enables the code fragment to compile?

- A. Replace line n1 with: `IntFunction inFu = x -> y -> x*y;`
- B. Replace line n1 with: `IntFunction inFu = x -> y -> x*y;`
- C. Replace line n1 with: `BiFunction inFu = x -> y -> x*y;`
- D. Replace line n2 with: `IntStream newStream = stream.map(inFu.applyAsInt (10));`

Correct Answer: B

---

### QUESTION 3

Given:

```
class UserException extends Exception { } class AgeOutOfLimitException extends UserException { }
```

and the code fragment:

```
class App { public void doRegister(String name, int age)
```

```
    throws UserException, AgeOutOfLimitException { if (name.length () = 60) { throw new AgeOutOfLimitException (); }  
    else { System.out.println("User is registered.");
```

```
    } } public static void main(String[ ] args) throws UserException {
```

```
    App t = new App (); t.doRegister("Mathew", 60); } }
```

What is the result?

- A. User is registered.
- B. An `AgeOutOfLimitException` is thrown.
- C. A `UserException` is thrown.
- D. A compilation error occurs in the main method.

Correct Answer: B

---

**QUESTION 4**

Given the code fragment:

```
ProductCode<Number, Integer> c1 = new ProductCode<Number, Integer>(); /* c1
instantiation */
ProductCode<Number, String> c2 = new ProductCode<Number, String>(); /* c2
instantiation */
```

You have been asked to define the ProductCode class. The definition of the ProductCode class must allow c1 instantiation to succeed and cause a compilation error on c2 instantiation.

Which definition of ProductCode meets the requirement?

A.

B.

```
class ProductCode<T, S<Integer>> {
    T c1;
    S c2;
}
```

```
class ProductCode<T, S extends T> {
    T c1;
    S c2;
}
```

C. 

```
class ProductCode<T, S> {
    T c1;
    S c2;
}
```

D. 

```
class ProductCode<T, S super T> {
    T c1;
    S c2;
}
```

C. D.

Correct Answer: B

---

**QUESTION 5**

Given that these files exist and are accessible:

```
/company/emp/info.txt
/company/emp/benefits/b1.txt
```

and given the code fragment:

```
// line n1  
stream.forEach(s -> System.out.print(s));
```

Which code fragment can be inserted at line n1 to enable the code to print only /company/emp?

- A. Stream stream = Files.list (Paths.get ("/company"));
- B. Stream stream = Files.find( Paths.get ("/company"), 1, (p,b) -andgt; b.isDirectory (), FileVisitOption.FOLLOW\_LINKS);
- C. Stream stream = Files.walk (Paths.get ("/company"));
- D. Stream stream = Files.list (Paths.get ("/company/emp"));

Correct Answer: B

[Latest 1Z0-809 Dumps](#)

[1Z0-809 PDF Dumps](#)

[1Z0-809 VCE Dumps](#)