

# 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.pass2lead.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 the code fragment:

```
UnaryOperator uo1 = s -> s*2; line n1
```

```
List loanValues = Arrays.asList(1000.0, 2000.0);
```

```
loanValues.stream()
```

```
.filter(lv -> lv >= 1500)
```

```
.map(lv -> uo1.apply(lv))
```

```
.forEach(s -> System.out.print(s + " "));
```

What is the result?

- A. 4000.0
- B. 4000
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: D

### QUESTION 3

Given:

```
class Block {  
    String color;  
    int size;  
    Block(int size, String color) {  
        this.size = size;  
        this.color = color;  
    }  
}
```

and the code fragment:

```
List<Block> blocks = new ArrayList<>();  
blocks.add(new Block(10, "Green"));  
blocks.add(new Block(7, "Red"));  
blocks.add(new Block(12, "Blue"));  
Collections.sort(blocks, new ColorSorter());
```

Which definition of the ColorSorter class sorts the blocks list?

- A. 

```
class ColorSorter implements Comparable<Block> {
    public boolean compare(Block o1, Block o2) {
        return o1.color.equals(o2.color);
    }
}
```
- B. 

```
class ColorSorter implements Comparable<Block> {
    public int compareTo(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}
```
- C. 

```
class ColorSorter implements Comparator<Block> {
    public int compare(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}
```
- D. 

```
class ColorSorter implements Comparator<Block> {
    public boolean compare(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}
```

A. B. C. D.

Correct Answer: C

#### QUESTION 4

Given the code fragments:

```
class ThreadRunner implements Runnable {
    public void run () { System.out.print ("Runnable"); }
}
```

```
class ThreadCaller implements Callable {
    Public String call () throws Exception {return "Callable"; }
}
```

and

```
ExecutorService es = Executors.newCachedThreadPool ();
```

```
Runnable r1 = new ThreadRunner ();
```

```
Callable c1 = new ThreadCaller ();
```

```
// line n1
```

```
es.shutdown();
```

Which code fragment can be inserted at line n1 to start r1 and c1 threads?

A. Future f1 = (Future) es.submit (r1); es.execute (c1);

B. es.execute (r1); Future f1 = es.execute (c1) ;

C. Future f1 = (Future) es.execute(r1); Future f2 = (Future) es.execute(c1);

D. es.submit(r1); Future f1 = es.submit (c1);

Correct Answer: D

## QUESTION 5

Given the definition of the Employee class: and this code fragment:

```
class Employee {
    String dept, name;
    public Employee (String d, String n) {
        dept = d;
        name = n;
    }
    public String toString() {
        return getDept () + ":" + getName ();
    }
    public String getDept () { return dept; }
    public String getName () { return name; }
}
```

```
List<Employee> emps = Arrays.asList(new Employee("sales", "Ada"),
    new Employee("sales", "Bob"),
    new Employee("hr", "Bob"),
    new Employee("hr", "Eva"));
Stream<Employee> s = emps.stream()
    .sorted(Comparator.comparing((Employee e) -> e.getDept())
        .thenComparing((Employee e) -> e.getName()));
List<Employee> eSorted = s.collect(Collectors.toList());
System.out.println(eSorted);
```

What is the result?

A. [sales:Ada, hr:Bob, sales:Bob, hr:Eva]

B. [Ada:sales, Bob:sales, Bob:hr, Eva:hr]

C. [hr:Eva, hr:Bob, sales:Bob, sales:Ada]

D. [hr:Bob, hr:Eva, sales:Ada, sales:Bob]

Correct Answer: A

[1Z0-809 Practice Test](#)

[1Z0-809 Exam Questions](#)

[1Z0-809 Braindumps](#)