

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.leads4pass.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

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.

Correct Answer: BD

Reference: <https://stackoverflow.com/questions/49476559/java-9-error-not-in-a-module-on-the-modulesource-path>

QUESTION 2

Given: Which statement is equivalent to line 1?

```
import java.util.List;
import java.util.function.BinaryOperator;
public class Main {
    public static void main(String... args) {
        List<Employee> list = List.of(new Employee("John", 80000.0), new Employee("Scott",
90000.0));
        double starts = 0.0;
        double ratio = 1.0;
        BinaryOperator<Double> bo = (a, b) -> a + b;
double totalSalary = list.stream().map(e -> e.getSalary() * ratio).reduce(starts, bo);
// line 1
        System.out.println("Total salary = " + totalSalary);
    }
}

class Employee {
    String name;
    double salary;
    public Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }
    public String getName() { return name; }
    public double getSalary() { return salary; }
}
```

- A. `double totalSalary = list.stream().map(e -> e.getSalary() * ratio).reduce(bo).ifPresent (p -> p.doubleValue());`
- B. `double totalSalary = list.stream().mapToDouble(e -> e.getSalary() * ratio).sum;`
- C. `double totalSalary = list.stream().map(Employee::getSalary * ratio).reduce(bo).orElse(0.0);`
- D. `double totalSalary = list.stream().mapToDouble(e -> e.getSalary() * ratio).reduce(starts, bo);`

Correct Answer: C

QUESTION 3

A bookstore's sales are represented by a list of Sale objects populated with the name of the customer and the books they purchased.

```
public class Sale {
    private String customer;
    private List items;
    // constructor, setters and getters not shown
}

public class Book {
    private String name;
```

```
private double price;
```

```
// constructor, setters and getters not shown
```

```
}
```

Given a list of Sale objects, tList, which code fragment creates a list of total sales for each customer in

ascending order?

- ```
A. List<String> totalByUser = tList.stream()
 .collect(flatMapping(t -> t.getItems().stream(),
 groupingBy(Sale::getCustomer,
 summingDouble(Book::getPrice))))
 .entrySet().stream()
 .sorted(Comparator.comparing(Entry::getValue))
 .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));
```
- ```
B. List<String> totalByUser = tList.stream()
    .collect(groupingBy(Sale::getCustomer,
        flatMapping(t -> t.getItems().stream(),
            summingDouble(Book::getPrice))))
    .sorted(Comparator.comparing(Entry::getValue))
    .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));
```
- ```
C. List<String> totalByUser = tList.stream()
 .collect(groupingBy(Sale::getCustomer,
 flatMapping(t -> t.getItems().stream(),
 summingDouble(Book::getPrice))))
 .entrySet().stream()
 .sorted(Comparator.comparing(Entry::getValue))
 .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));
```
- ```
D. List<String> totalByUser = tList.stream()
    .collect(flatMapping(t -> t.getItems().stream(),
        groupingBy(Sale::getCustomer,
            summingDouble(Book::getPrice))))
    .sorted(Comparator.comparing(Entry::getValue))
    .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

QUESTION 4

Given: Which one is correct?

```
public class Main {
    public static void main(String[] args) {
        Thread t1 = new Thread(new MyThread());
        Thread t2 = new Thread(new MyThread());
        Thread t3 = new Thread(new MyThread());

        t1.start();
        t2.run();
        t3.start();

        t1.start();
    }
}
class MyThread implements Runnable {
    public void run() {
        System.out.println("Running.");
    }
}
```

- A. An `IllegalThreadStateException` is thrown at run time.
- B. Three threads are created.
- C. The compilation fails.
- D. Four threads are created.

Correct Answer: A

CPD Time: 0.10 sec(s), Memory: 32.100 Kibyte(s)

```
Running.
Running.
Running.
```

```
Exception in thread "main" java.lang.IllegalThreadStateException
    at java.base/java.lang.Thread.start(Thread.java:794)
    at Main.main(Main.java:12)
```

QUESTION 5

Given:

```
@Target (ElementType.METHOD)
@Retention (RetentionPolicy.RUNTIME)
public @interface AuthorInfo {
    String author() default "";
    String date();
    String[] comments() default {};
}
```

Which two are correct? (Choose two.)

- A.

```
@AuthorInfo (date="1-1-2020", comments={ null })
public class Hello {
    public void func() {}
}
```
- B.

```
public class Hello {
    @AuthorInfo (date="1-1-2020. comments="Hello")
    public void func() {}
}
```
- C.

```
public class Hello {
    @AuthorInfo
    public void func() {}
}
```
- D.

```
@AuthorInfo (date="1-1-2020")
public class Hello {
    public void func() {}
}
```
- E.

```
public class Hello {
    @AuthorInfo (date="1-1-2020", author="Gandhi", comments={ "world" })
    public void func () {}
}
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

Correct Answer: CD