



# 1Z0-816<sup>Q&As</sup>

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.pass4itsure.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**

```
var numbers = List.of(0,1,2,3,4,5,6,7,8,9);
```

You want to calculate the average of numbers.

Which two codes will accomplish this? (Choose two.)

- A. `double avg = numbers.stream().parallel().averagingDouble(a -> a);`
- B. `double avg = numbers.parallelStream().mapToInt (m -> m).average().getAsDouble();`
- C. `double avg = numbers.stream().mapToInt (i -> i).average().parallel();`
- D. `double avg = numbers.stream().average().getAsDouble();`
- E. `double avg = numbers.stream().collect(Collectors.averagingDouble(n -> n));`

Correct Answer: BD

```
1
2 import java.io.*;
3 import java.util.*;
4 class Hello {
5     public static void main(String[] args) {
6
7         var numbers = List.of(0,1,2,3,4,5,6,7,8,9);
8         double avg = numbers.parallelStream().mapToInt (m -> m).average().getAsDouble();
9
10    }
11 }
```

**QUESTION 2**

Given:

```
1. public class Test {
2.     private static class Greet {
3.         private void print() {
4.             System.out.println("Hello World");
5.         }
6.     }
7.     public static void main(String[] args) {
8.         Test.Greet i = new Greet();
9.         i.print();
10.    }
11. }
```

What is the result?



- A. The compilation fails at line 9.
- B. The compilation fails at line 2.
- C. Hello World
- D. The compilation fails at line 8.

Correct Answer: C

### QUESTION 3

Given: Which option should you choose to enable the code to print Something happened?

```
public class Test {
    public static void doThings() throws GeneralException {
        try {
            throw new RuntimeException("Someting happened");
        } catch (Exception e) {
            throw new SpecificException(e.getMessage());
        }
    }
    public static void main(String args[]) {
        try{
            Test.doThings();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
class GeneralException /* line 1 */ {
    public GeneralException(String s) { super(s); }
}
class SpecificException /* line 2 */ {
    public SpecificException(String s) { super(s); }
}
```

- A. Add extends GeneralException on line 1. Add extends Exception on line 2.
- B. Add extends SpecificException on line 1. Add extends GeneralException on line 2.
- C. Add extends Exception on line 1. Add extends Exception on line 2.
- D. Add extends Exception on line 1. Add extends GeneralException on line 2.

Correct Answer: D



```
1 import java.util.*;
2 import java.io.*;
3 import java.lang.Thread;
4 import java.util.ArrayList;
5 import java.util.LinkedList;
6 import java.util.List;
7
8 public class Test {
9
10     public static void doThings() throws GeneralException {
11         try{
12             throw new RuntimeException("Something happened");
13         } catch (Exception e) {
14             throw new SpecificException (e.getMessage());
15         }
16     }
17 }
18
19 public static void main(String args[]) {
20     try{
21         Test.doThings();
22     }catch (Exception e) {
23         System.out.println(e.getMessage());
24     }
25 }
26 class GeneralException extends Exception {
27     public GeneralException(String s) { super(s); }
28 }
29 class SpecificException extends GeneralException {
30     public SpecificException(String s) { super(s);}
31 }
32 }
```

#### QUESTION 4

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 5

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=tandrc=jandq=andesrc=sandsource=webandcd=4andved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxABandurl=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%2Fletter%2Fen_US%2Fdb2s2e90.pdfandusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec)

[sa=tandrc=jandq=andesrc=sandsource=webandcd=4andved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxABandurl=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%](https://www.google.com/url?sa=tandrc=jandq=andesrc=sandsource=webandcd=4andved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxABandurl=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%2Fletter%2Fen_US%2Fdb2s2e90.pdfandusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec)

[2Fletter%2Fen\\_US%2Fdb2s2e90.pdfandusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec](https://www.google.com/url?sa=tandrc=jandq=andesrc=sandsource=webandcd=4andved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxABandurl=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%2Fletter%2Fen_US%2Fdb2s2e90.pdfandusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec)

[2Fletter%2Fen\\_US%2Fdb2s2e90.pdfandusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec](https://www.google.com/url?sa=tandrc=jandq=andesrc=sandsource=webandcd=4andved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxABandurl=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%2Fletter%2Fen_US%2Fdb2s2e90.pdfandusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec)

## QUESTION 6

Given:

```
class CustomType<T> {
    public <T> int count(T[] anArray, T element) {
        int count = 0;
        for(T e : anArray) {
            if (e.equals(element)) ++count;
        }
        return count;
    }
}
```

and

```
public class Test extends CustomType {
    public static void main(String[] args) {
        String[] words = {"banana", "orange", "apple", "lemon"};
        Integer[] numbers = {1, 2, 3, 4, 5};
        CustomType type = new CustomType();
        CustomType<String> stringType = new CustomType<>();
        System.out.println(stringType.count(words, "apple"));
        System.out.println(type.count(words, "apple"));
        System.out.println(type.count (numbers, 3));
    }
}
```

What is the result?



- A. A NullPointerException is thrown at run time.
- B. The compilation fails.
- C. 1 Null null
- D. 1
- E. A ClassCastException is thrown at run time.

Correct Answer: B

Console 4

```
Error: Could not find or load main class CustomType
Caused by: java.lang.ClassNotFoundException: CustomType
```

---

#### QUESTION 7

Given:

```
LocalDate d1 = LocalDate.of(1997,2,7); DateTimeFormatter dtf = DateTimeFormatter.ofPattern( /*insert code here*/ );
System.out.println(dtf.format (d1));
```

 Which pattern formats the date as Friday 7th of February 1997?

- A. "eeee dd+"th of"+ MMM yyyy"
- B. "eeee dd\\'th of\\' MMM yyyy"
- C. "eeee d+"th of"+ MMMM yyyy"
- D. "eeee d\\'th of\\' MMMM yyyy"

Correct Answer: B

Reference: [https://books.google.com.pk/books?id=PmiO65T9hF0Candpg=PA385andlpg=PA385anddq=java+pattern+formats+eeee+d%2Bth+of%2B+MMMM+yyyyandsource=blandots=IjN\\_-AnWQjandsig=ACfU3U2RjF7iuK3t\\_SKARwLSaak9xxV09Aandhl=enandsa=Xandved=2ahUKEwi4m6LL3vLoAhVgTRUIHURpC38Q6AEwDHoECBQQAQ#v=onepageandq=java%20pattern%20formats%20eeee%20d%2Bth%20of%2B%20MMMM%20yyyyandf=false](https://books.google.com.pk/books?id=PmiO65T9hF0Candpg=PA385andlpg=PA385anddq=java+pattern+formats+eeee+d%2Bth+of%2B+MMMM+yyyyandsource=blandots=IjN_-AnWQjandsig=ACfU3U2RjF7iuK3t_SKARwLSaak9xxV09Aandhl=enandsa=Xandved=2ahUKEwi4m6LL3vLoAhVgTRUIHURpC38Q6AEwDHoECBQQAQ#v=onepageandq=java%20pattern%20formats%20eeee%20d%2Bth%20of%2B%20MMMM%20yyyyandf=false)

---

#### QUESTION 8

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 9

Given:

```
public class Main {
    public static void main(String[] args) {
        try (BufferedReader br = new BufferedReader(new InputStreamReader(System.in));) {
            String input = br.readLine();
            System.out.println ("Input String was: " + input);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Which is true?

- A. System.out is the standard output stream. The stream is open only when System.out is called.
- B. System.in cannot reassign the other stream.
- C. System.out is an instance of java.io.OutputStream by default.
- D. System.in is the standard input stream. The stream is already open.

Correct Answer: D

Reference: <https://www.geeksforgeeks.org/java-lang-system-class-java/>

---

### QUESTION 10

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

Execute



#### Result

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

orangebanana\lemon

#### QUESTION 11

Which two are successful examples of autoboxing? (Choose two.)

- A. String a = "A";
- B. Integer e = 5;
- C. Float g = Float.valueOf(null);
- D. Double d = 4;
- E. Long c = 23L;
- F. Float f = 6.0;

Correct Answer: AB



### QUESTION 12

Given:

```
List list1 = new ArrayList();
```

```
list1.add("A");
```

```
list1.add("B");
```

```
List list2 = List.copyOf(list1);
```

```
list2.add("C");
```

```
List list3 = List.of(list1, list2);
```

```
System.out.println(list3);
```

What is the result?

A. [[A, B],[A, B]]

B. An exception is thrown at run time.

C. [[A, B], [A, B, C]]

D. [[A, B, C], [A, B, C]]

Correct Answer: B



```
11
12 public class Main {
13     public static void main(String[] args) {
14
15         List<String> list1 = new ArrayList<>();
16         list1.add("A");
17         list1.add("B");
18         List list2 = List.copyOf(list1);
19         list2.add("C");
20         List<List<String>> list3 = List.of(list1, list2);
21         System.out.println(list3);
22     }
23
24 }
25
```

Execute Mode, Version, Inputs & Arguments

JDK 110.4  Interactive Stdin Inputs

CommandLine Arguments

Execute

### Result

CPU Time: 0.16 sec(s), Memory: 32128 kilobyte(s)

```
Exception in thread "main" java.lang.UnsupportedOperationException
at java.base/java.util.ImmutableCollections.uoe(ImmutableCollections.java:71)
at java.base/java.util.ImmutableCollections$AbstractImmutableCollection.add(ImmutableCollections.java:75)
at Main.main(Main.java:19)
```

### QUESTION 13

Given:

```
1. void insertionSort(int values[]) {
2.     int n = values.length;
3.     for (int j = 1; j < n; j++) {
4.         int tmp = values[j];
5.         int i = j - 1;
6.         while ( (i > -1) && (values[i] > tmp) ) {
7.             values[i + 1] = values[i];
8.             i--;
9.         }
10.        values[i + 1] = tmp;
11.    }
12. }
```

After which line can we insert `assert i < 0 || values[i]`