

**100%** Money Back  
**Guarantee**

**Vendor:**Oracle

**Exam Code:**1Z0-819

**Exam Name:**Java SE 11 Developer

**Version:**Demo

## QUESTION 1

Given:

```
package b;
public class Person {
    protected Person() { //line 1
    }
}
```

and

```
package a;
import b.Person;
public class Main { //line 2
    public static void main(String[] args) {
        Person person = new Person(); //line 3
    }
}
```

Which two allow a.Main to allocate a new Person? (Choose two.)

- A. In Line 1, change the access modifier to privateprivate Person() {
- B. In Line 1, change the access modifier to publicpublic Person() {
- C. In Line 2, add extends Person to the Main classpublic class Main extends Person {and change Line 3 to create a new Main objectPerson person = new Main();
- D. In Line 2, change the access modifier to protectedprotected class Main {
- E. In Line 1, remove the access modifierPerson() {

Correct Answer: BC

---

## QUESTION 2

Given:

```

public class FunctionalInterfaceTest {
    public static void main(String[] args) {
        List fruits = Arrays.asList("apple", "orange", "banana");
        Consumer<String> c = System.out::print;
        Consumer<String> output = c.andThen(x -> System.out.println(": " + x.toUpperCase
    ));
        fruits.forEach(output);
    }
}

```

What is the output?

- A. :APPLE:ORANGE:BANANA appleorangebanana
- B. :APPLE:ORANGE:BANANA
- C. APPLE:apple ORANGE:orange BANANA:banana
- D. appleorangebanana :APPLE:ORANGE:BANANA
- E. apple:APPLE orange:ORANGE banana:BANANA

Correct Answer: E

### QUESTION 3

Given the code fragment:

```

public class FizzBuzz {
    public static String convert(int x) {
        if (x % 15 == 0) return "FizzBuzz";
        else if (x % 3 == 0) return "Fizz";
        else if (x % 5 == 0) return "Buzz";
        else return Integer.toString(x);
    }

    public static void main(String[] args) {
        for (int i = 1; i < 101; i++) {
            System.out.println(convert(i));
        }
    }
}

```

Which code fragment replaces the for statement?

- A. IntStream.rangeClosed(1, 100).map(FizzBuzz::convert).forEach(System.out::println);
- B. IntStream.ranged, 100).map(FizzBuzz::convert).forEach(System.out::println);
- C. intstream.rangeclosed(1, 100).mapToObj{FizzBuzz::convert}.forEach(System.out::println);
- D. IntStream.range(1, 100).mapToObj(FizzBuzz::convert).forEach(System.out::println);

Correct Answer: A

---

#### QUESTION 4

Given:

```
var data = new ArrayList();  
data.add("Peter");  
data.add(30);  
data.add("Market Road");  
data.set(1, 25);  
data.remove(2);  
data.set(3, 1000L);  
System.out.print(data);
```

What is the output?

- A. [Market Road, 1000]
- B. [Peter, 30, Market Road]
- C. [Peter, 25, null, 1000]
- D. An exception is thrown at run time.

Correct Answer: D

```
Console 1 ✖  
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index 3 out of bounds for length 2  
    at java.base/jdk.internal.util.Preconditions.outOfBounds(Preconditions.java:64)  
    at java.base/jdk.internal.util.Preconditions.outOfBoundsCheckIndex(Preconditions.java:70)  
    at java.base/jdk.internal.util.Preconditions.checkIndex(Preconditions.java:248)  
    at java.base/java.util.Objects.checkIndex(Objects.java:372)  
    at java.base/java.util.ArrayList.set(ArrayList.java:472)  
    at abc.main(abc.java:13)  
Completed with exit code: 1
```

---

#### QUESTION 5

Given:

```
public class Tester {  
    public static void main(String[] args) {  
        StringBuilder sb = new StringBuilder(5);  
        sb.append("HOWDY");  
        sb.insert(0, ' ');  
        sb.replace(3, 5, "LL");  
        sb.insert(6, "COW");  
        sb.delete(2, 7);  
        System.out.println(sb.length());  
    }  
}
```

What is the result?

- A. 4
- B. 3
- C. An exception is thrown at runtime.
- D. 5

Correct Answer: A

```
1 import java.lang.StringBuilder;
2 public class Tester {
3     public static void main(String[] args) {
4         StringBuilder sb = new StringBuilder(5);
5         sb.append("HOWDY");
6         sb.insert(0, ' ');
7         sb.replace(3, 5, "LL");
8         sb.insert(6, "COW");
9         sb.delete(2, 7);
10        System.out.println(sb.length());
11    }
12 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

### Result

CPU Time: 0.12 sec(s), Memory: 30392 kilobyte(s)

4

interface, text, application, chat or text message

### QUESTION 6

Which two statements are correct about modules in Java? (Choose two.)

- A. module-info.java cannot be empty
- B. module-info.java can be placed in any folder inside module-path
- C. By default, modules can access each other as long as they run in the same folder
- D. A module must be declared in module-info.java file
- E. java.base exports all of the Java platforms core packages

Correct Answer: DE

---

### QUESTION 7

Given the code fragment:

```
Path source = Paths.get("/repo/a/a.txt");
```

```
Path destination = Paths.get("/repo");
```

```
Files.move(source, destination); // line 1
```

```
Files.delete (source); // line 2
```

Assuming the source file and destination folder exist, what is the result?

- A. A `java.nio.file.FileAlreadyExistsException` is thrown on line 1.
- B. A `java.nio.file.NoSuchFileException` is thrown on line 2.
- C. A copy of `/repo/a/a.txt` is moved to the `/repo` directory and `/repo/a/a.txt` is deleted.
- D. `a.txt` is renamed `repo`.

Correct Answer: C

---

### QUESTION 8

Consider this method declaration:

```
void setSessionUser(Connection conn, String user) throws SQLException {  
    Statement stmt = conn.createStatement();  
    String sql = <EXPRESSION>;  
    stmt .execute();  
}
```

"SET SESSION AUTHORIZATION " + user

"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. 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?sa=t&drct=j&ndq=andesrc=s&ndsource=web&ndcd=4&ndved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxAB&ndurl=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%2Fletter%2Fen\\_US%2Fdb2s2e90.pdf&ndusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec](https://www.google.com/url?sa=t&drct=j&ndq=andesrc=s&ndsource=web&ndcd=4&ndved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxAB&ndurl=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%2Fletter%2Fen_US%2Fdb2s2e90.pdf&ndusg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec)

---

## QUESTION 9

Given:

```
class Super {
    final int num; // line n1
    public Super(int num) {
        this.num = num;
    }
    final void method() {
        System.out.println("Output from Super");
    }
}
class Sub extends Super {
    int num; // line n2
    Sub(short num) { // line n3
        super(num);
    }
    protected void method() { // line n4
        System.out.println("Output from Sub");
    }
}
```

Which line of code results in a compilation error?

- A. line n1
- B. line n3
- C. line n2
- D. line n4

Correct Answer: D

---

## QUESTION 10

Given:

```
class MyPersistenceData {
    String str;
    private void methodA() {
        System.out.println("methodA");
    }
}
```



You want to implement the java.io.Serializable interface to the MyPersisteneData class. Which method should be overridden?

- A. The readExternal and writeExternal method
- B. The readExternal method
- C. The writeExternal method
- D. nothing

Correct Answer: A

---

#### QUESTION 11

Given: Which annotation should be used to remove warnings from compilation?

```
public class Main {
    public static void main(String[] args) {
        List l = new ArrayList();
        l.add("hello");
        l.add("world");
        print(l);
    }
    private static void print(List<String>... args) {
        for (List<String> str : args) {
            System.out.println (str);
        }
    }
}
```

- A. @SuppressWarnings on the main and print methods
- B. @SuppressWarnings("unchecked") on main and @SafeVarargs on the print method
- C. @SuppressWarnings("rawtypes") on main and @SafeVarargs on the print method
- D. @SuppressWarnings("all") on the main and print methods

Correct Answer: B

```
13 @SuppressWarnings("unchecked")
14 public class Main {
15
16     public static void main(String[] args) {
17
18         List l = new ArrayList();
19         l.add("Hello");
20         l.add("world");
21         print(l);
22
23     }
24
25     private static void print(List<String>... args) {
26         for (List<String> str : args) {
27             System.out.println (str);
28
29         }
30     }
31     @SafeVarargs
32 }
```

---

## QUESTION 12

Given: What is the result?

Given:

```
public class ExSuper extends Exception {
    private final int eCode;
    public ExSuper(int eCode, Throwable cause) {
        super(cause);
        this.eCode = eCode;
    }

    public ExSuper(int eCode, String msg, Throwable cause) {
        super(msg, cause);
        this.eCode = eCode;
    }
    public String getMessage() {
        return this.eCode+": "+super.getMessage()+"-"+this.getCause().getMessage();
    }
}

public class ExSub extends ExSuper {
    public ExSub(int eCode, String msg, Throwable cause)
        { super(eCode, msg, cause); }
}
```

and the code fragment:

```
try {
    String param1 = "Oracle";
    if (param1.equalsIgnoreCase("oracle")) {
        throw new ExSub(9001, "APPLICATION ERROR-9001", new FileNotFoundException ("MyFile.txt"));
    }
    throw new ExSuper(9001, new FileNotFoundException ("MyFile.txt")); // Line 1
} catch (ExSuper ex) {
    System.out.println(ex.getMessage());
}
```

- A. 9001: java.io.FileNotFoundException: MyFile.txt-MyFile.txt
- B. 9001: APPLICATION ERROR-9001-MyFile.txt 9001: java.io.FileNotFoundException: MyFile.txt-MyFile.txt
- C. 9001: APPLICATION ERROR-9001-MyFile.txt
- D. Compilation fails at Line 1.

Correct Answer: C