

# ***KillTest***

Higher Quality, Better Service!



## **Q&A**

<http://www.killtest.com>

We offer free update service for one year.

**Exam : 1Z0-815**

**Title : Java SE 11 Programmer I**

**Version : DEMO**

1.Given:

```
public class A {  
    private boolean checkValue(int val) {  
        return true;  
    }  
}
```

and

```
public class B extends A {  
    public int modifyVal(int val) {  
        if(checkValue(val)) {  
            return val;  
        } else {  
            return 0;  
        }  
    }  
    public static void Main(String[] args) {  
        B b = new B();  
        System.out.println(b.modifyVal(10));  
    }  
}
```

What is the result?

- A. nothing
- B. It fails to compile.
- C. 0
- D. A java.lang.IllegalArgumentException is thrown.
- E. 10

**Answer: B**

**Explanation:**

```
1 public class A {
2     private boolean checkValue(int val) {
3         return true;
4     }
5 }
6 and
7 public class B extends A {
8     public int modifyVal(int val) {
9         if(checkValue(val)) {
10            return val;
11        } else {
12            return 0;
13        }
14    }
15    public static void Main(String[] args) {
16        B b = new B();
17        system.out.println(b.modfiyVal (10));
18    }
19 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

## Result

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

```
/A.java:6: error: class, interface, or enum expected
and
^
1 error
```

2.Given:

```
public interface API { //line 1
    public void checkValue(Object value)
        throws IllegalArgumentException; //line 2
    public boolean isValueANumber(Object val) {
        if(val instanceof Number) {
            return true;
        }else {
            try {
                Double.parseDouble(val.toString());
                return true;
            }catch (NumberFormatException ex) {
                return false;
            }
        }
    }
}
```

Which two changes need to be made to make this class compile? (Choose two.)

- A. Change Line 1 to an abstract class:public abstract class API {
- B. Change Line 2 access modifier to protected:protected void checkValue(Object value)throws IllegalArgumentException;
- C. Change Line 1 to a class:public class API {
- D. Change Line 1 to extend java.lang.AutoCloseable:public interface API extends AutoCloseable {
- E. Change Line 2 to an abstract method:public abstract void checkValue(Object value)throws IllegalArgumentException;

**Answer:** A,E

3.Which two modules include APIs in the Java SE Specification? (Choose two.)

- A. java.logging
- B. java.desktop
- C. javafx D. jdk.httpserver
- E. jdk.jartool

**Answer:** A,C

Explanation: <https://docs.oracle.com/javase/9/docs/api/overview-summary.html>

4.Given:

```
public class Test{
    private int num = 1;
    private int div = 0;

    public void divide() {
        try {
            num = num / div;
            System.out.print("Exception");
        }
        catch(ArithmeticException ae) { num = 100; }
        catch(Exception e) { num = 200; }
        finally { num = 300; }
        System.out.print(num);
    }
    public static void main(String args[])
    {
        Test test = new Test();
        test.divide();
    }
}
```

What is the output?

- A. 300
- B. Exception
- C. 200
- D. 100

**Answer:** A

**Explanation:**

```
1- public class Test{
2     private int num = 1;
3     private int div = 0;
4
5     public void divide() {
6         try {
7             num = num / div;
8             System.out.print("Exception");
9         }
10        catch(ArithmeticException ae) { num = 100; }
11        catch(Exception e) { num = 200; }
12        finally { num = 300; }
13        System.out.print(num);
14    }
15    public static void main(String args[])
16    {
17        Test test = new Test();
18        test.divide();
19    }
20 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.15 sec(s), Memory: 32484 kilobyte(s)

300

5.Which two statements are true about the modular JDK? (Choose two.)

- A. The foundational APIs of the Java SE Platform are found in the java.base module.
- B. An application must be structured as modules in order to run on the modular JDK.
- C. It is possible but undesirable to configure modules' exports from the command line.
- D. APIs are deprecated more aggressively because the JDK has been modularized.

**Answer:** A,B