

# Oracle

## Exam Questions 1Z0-819

Java SE 11 Developer



### NEW QUESTION 1

Given:

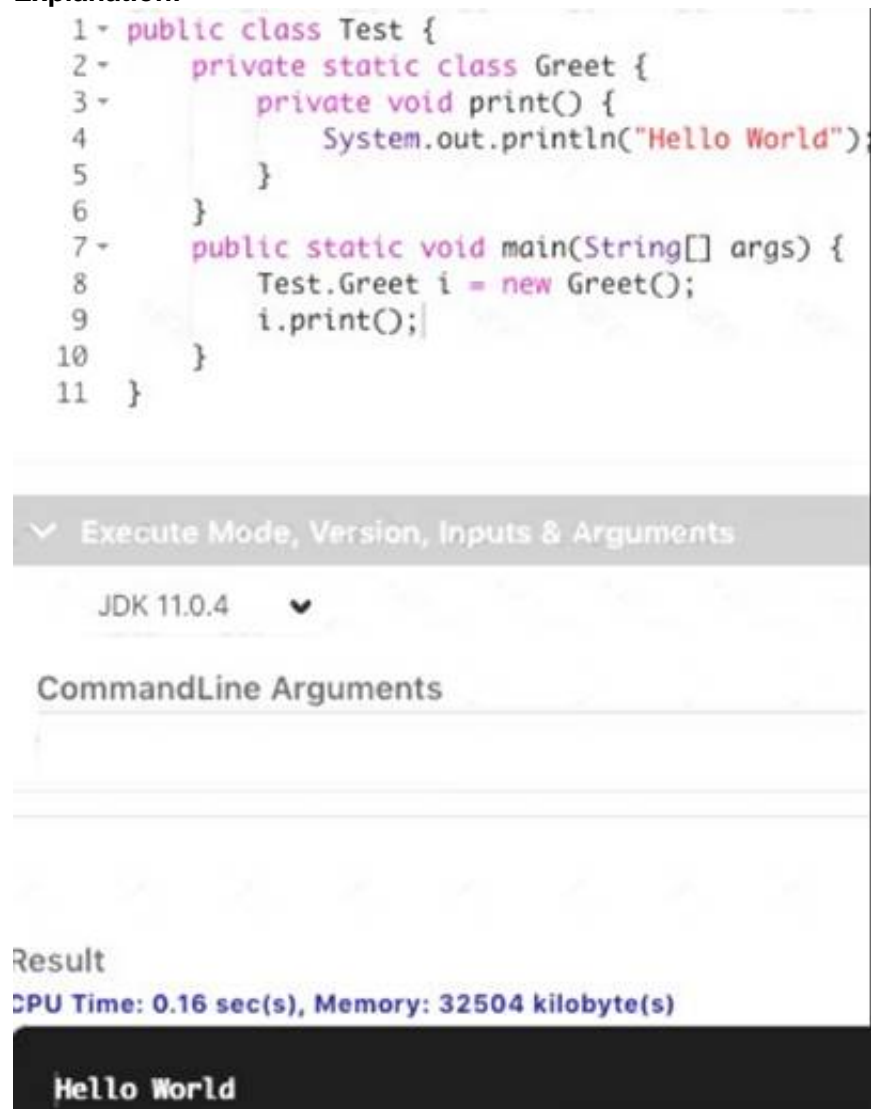
```
1. public class Test {  
2.     private static class Greet {  
3.         private void print() {  
4.             System.out.println("Hello World");  
5.         }  
6.     }  
7.     public static void main(String[] args) {  
8.         Test.Greet i = new Greet();  
9.         i.print();  
10. 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

**Answer: C**

**Explanation:**



```
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 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32504 kilobyte(s)

Hello World

### NEW QUESTION 2

Given:

```
public interface Builder {  
    public A build(String str);  
}
```

and

```
public class BuilderImpl implements Builder {  
    @Override  
    public B build(String str) {  
        return new B(str);  
    }  
}
```

Assuming that this code compiles correctly, which three statements are true? (Choose three.)

- A. B cannot be abstract.
- B. B is a subtype of A.
- C. A cannot be abstract.
- D. A cannot be final.
- E. B cannot be final.
- F. A is a subtype of B.

**Answer:** ABD

### NEW QUESTION 3

Which interface in the java.util.function package will return a void return type?

- A. Supplier
- B. Predicate
- C. Function
- D. Consumer

**Answer:** D

### NEW QUESTION 4

Given:

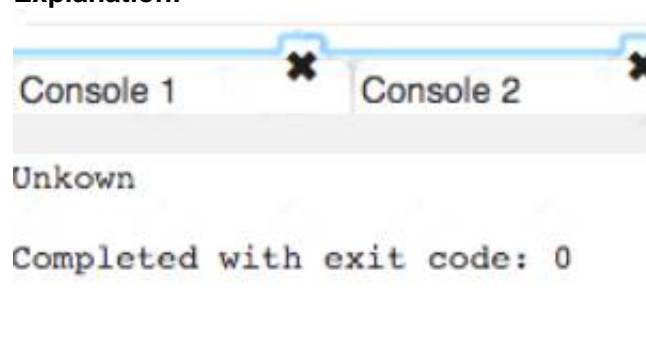
```
import java.time.LocalDate;
import static java.time.DayOfWeek.*;
public class Main {
    public static void main(String[] args) {
        var today = LocalDate.now().with(TUESDAY).getDayOfWeek();
        switch(today) {
            case SUNDAY:
            case SATURDAY:
                System.out.println("Weekend");
                break;
            case MONDAY: FRIDAY:
                System.out.println("Working");
            default:
                System.out.println("Unknown");
        }
    }
}
```

What is the result?

- A. WorkingUnknown
- B. Unknown
- C. TuesdayUnknown
- D. The compilation fails.
- E. Tuesday
- F. Working

**Answer:** B

**Explanation:**



### NEW QUESTION 5

Which code fragment does a service use to load the service provider with a Print interface?

- A. private Print print = com.service.Provider.getInstance();
- B. private java.util.ServiceLoader<Print> loader = ServiceLoader.load(Print.class);
- C. private java.util.ServiceLoader<Print> loader = new java.util.ServiceLoader<>();
- D. private Print print = new com.service.Provider.PrintImpl();

**Answer:** B

### NEW QUESTION 6

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:** AC

#### NEW QUESTION 7

Examine this excerpt from the declaration of the java.se module:

```
module java.se {  
    ...  
    requires transitive java.sql;  
    ...  
}
```

What does the transitive modifier mean?

- A. Only a module that requires the java.se module is permitted to require the java.sql module.
- B. Any module that requires the java.se module does not need to require the java.sql module.
- C. Any module that attempts to require the java.se module actually requires the java.sql module instead.
- D. Any module that requires the java.sql module does not need to require the java.se module.

**Answer:** A

#### NEW QUESTION 8

Given:

```
public class MethodTest {  
    // line 1  
}
```

Which two method implementations are correct, when inserted independently in line 1? (Choose two.)

A.

```
public boolean methodD(int x) {  
    return x > 0;  
}
```

B.

```
public String methodB() {  
    System.out.println("methodB");  
}
```

C.

```
public char methodE (String msg) {  
    return msg;  
}
```

D.

```
public void methodC(int x) {  
    return ++x;  
}
```

E.

```
public void methodA() {  
    System.out.println("methodA");  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** AE

#### NEW QUESTION 9

Given:

```
int arr[][] = {{5,10},{8,12},{9,3}};
long count = Stream.of(arr)
    .flatMapToInt(IntStream::of)
    .map(n -> n + 1)
    .filter(n -> (n % 2 == 0))
    .peek(System.out::print)
    .count();

System.out.println(" " + count);
```

What is the result?

- A. 6910 3
- B. 10126 3
- C. 3
- D. 6104 3

**Answer: D**

**Explanation:**

```
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  import java.util.function.Consumer;
8  import java.util.stream.Stream;
9  import java.util.stream.IntStream;
10
11
12 - public class Main {
13
14 -     public static void main(String[] args) {
15         int arr[][] = {{5,10}, {8,12}, {9,3}};
16         long count = Stream.of(arr)
17             .flatMapToInt(IntStream::of)
18             .map(n -> n + 1)
19             .filter(n -> (n % 2 == 0))
20             .peek(System.out::print)
21             .count();
22         System.out.println(" " + count);
23     }
24 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.32 sec(s), Memory: 34220 kilobyte(s)

**6104 3**

**NEW QUESTION 10**

Given:



```
public class Tester {
    static class Person implements /* line 1 */ {
        private String name;
        Person(String name) { this.name = name; }
        /* line 2 */
    }
    public static void main(String[] args) {
        Person[] people = {new Person("Joe"),
                           new Person("Jane"),
                           new Person("John")};
        Arrays.sort(people);
        for(Person person: people) {
            System.out.println(person.name);
        }
    }
}
```

You want the code to produce this output:

John

Joe Jane

Which code fragment should be inserted on line 1 and line 2 to produce the output?

- A. Insert `Comparator<Person>` on line 1. Insert `public int compare(Person p1, Person p2) { return p1.name.compare(p2.name);}` on line 2.
- B. Insert `Comparator<Person>` on line 1. Insert `public int compareTo(Person person) { return person.name.compareTo(this.name);}` on line 2.
- C. Insert `Comparable<Person>` on line 1. Insert `public int compare(Person p1, Person p2) { return p1.name.compare(p2.name);}` on line 2.
- D. Insert `Comparator<Person>` on line 1. Insert `public int compare(Person person) { return person.name.compare(this.name);}` on line 2.

**Answer: B**

#### NEW QUESTION 10

Given:

```
public class Main {
    class Student {                                // line 1
        String classname;
        Student(String classname) {                // line 2
            this.classname = classname;
        }
    }
    public static void main(String[] args) {
        var student = new Student("Biology"); // line 3
    }
}
```

Which two independent changes will make the Main class compile? (Choose two.)

- A. Move the entire Student class declaration to a separate Java file, Student.java.
- B. Change line 2 to `public Student(String classname)`.
- C. Change line 1 to `public class Student {`.
- D. Change line 3 to `Student student = new Student("Biology");`.
- E. Change line 1 to `static class Student {`.

**Answer: BD**

**Explanation:**

```

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  import java.util.function.Consumer;
8  import java.util.stream.Stream;
9  import java.util.stream.IntStream;
10 import java.util.Optional;
11
12
13 - public class Main {
14 -     class Student {
15         String classname;
16 -     public Student (String classname) {
17         this.classname = classname;
18     }
19
20     }
21 -     public static void main (String[] args) {
22         var student = new Student ("Biology");
23     }
24 }

```

#### NEW QUESTION 12

Given:

```

public interface A {
    abstract void x();
}

```

and

```

public abstract class B /* position 1 */ {
    /* position 2 */
    public void x() { }
    public abstract void z();
}

```

and

```

public class C extends B implements A {
    /* position 3 */
}

```

Which code, when inserted at one or more marked positions, would allow classes B and C to compile?

- A. @Override // position 3 void x () {} // position 3 @Override // position 3 public void z() {} // position 3
- B. @Override // position 2 public void z() {} // position 3
- C. implements A // position 1 @Override // position 2
- D. public void z() {} // position 3

**Answer: B**

#### NEW QUESTION 17

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"

**Answer: B**

#### NEW QUESTION 20

Given:

```
public class Person {  
    private String name;  
    public void setName(String name) {  
        String title = "Dr. ";  
        name = title+name;  
    }  
    public String toString() {  
        return name;  
    }  
}
```

and

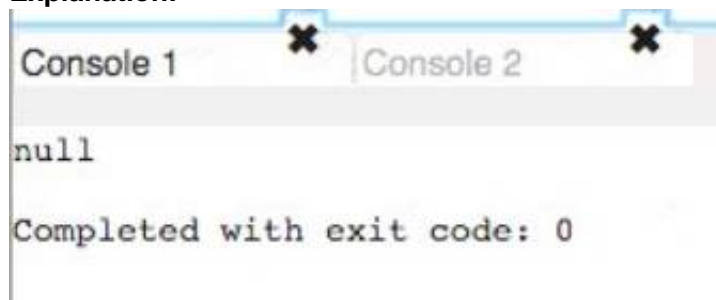
```
public class Test {  
    public static void main(String args[]) {  
        Person p = new Person();  
        p.setName("Who");  
        System.out.println(p);  
    }  
}
```

What is the result?

- A. D
- B. Who
- C. Dr
- D. Null
- E. An exception is thrown at runtime.
- F. null

**Answer:** D

**Explanation:**



```
Console 1 X Console 2 X  
null  
Completed with exit code: 0
```

#### NEW QUESTION 21

Given:

```
1. public class Secret {  
2.     String[] names;  
3.     public Secret(String[] names) {  
4.         this.names = names;  
5.     }  
6.     public String[] getNames() {  
7.         return names;  
8.     }  
9. }
```

Which three actions implement Java SE security guidelines? (Choose three.)

- A. Change line 7 to return names.clone();.
- B. Change line 4 to this.names = names.clone();.
- C. Change the getNames() method name to get\$Names().
- D. Change line 6 to public synchronized String[] getNames() {.
- E. Change line 2 to private final String[] names;.
- F. Change line 3 to private Secret(String[] names) {.
- G. Change line 2 to protected volatile String[] names;.

**Answer:** EFG

#### NEW QUESTION 25

Given:



```
public class Foo {  
    public void foo(Collection arg) {  
        System.out.println("Bonjour le monde!");  
    }  
}
```

and

```
public class Bar extends Foo {  
    public void foo(Collection arg) {  
        System.out.println("Hello world!");  
    }  
    public void foo(List arg) {  
        System.out.println("Hola Mundo!");  
    }  
}
```

and

```
Foo f1 = new Foo();  
Foo f2 = new Bar();  
Bar b1 = new Bar();  
List<String> li = new ArrayList<>();
```

Which three are correct? (Choose three.)

- A. b1.foo(li) prints Hello world!
- B. f1.foo(li) prints Bonjour le monde!
- C. f1.foo(li) prints Hello world!
- D. f1.foo(li) prints Hola Mundo!
- E. b1.foo(li) prints Bonjour le monde!
- F. f2.foo(li) prints Hola Mundo!
- G. f2.foo(li) prints Bonjour le monde!
- H. b1.foo(li) prints Hola Mundo!
- I. f2.foo(li) prints Hello world!

**Answer:** ABH

#### NEW QUESTION 26

Examine these module declarations:

```
module ServiceAPI {  
    exports com.example.api;  
}  
  
module ServiceProvider {  
    requires ServiceAPI;  
    provides com.example.api with com.myimpl.Impl;  
}  
  
module Consumer {  
    requires ServiceAPI;  
    uses com.example.api;  
}
```

Which two statements are correct? (Choose two.)

- A. The ServiceProvider module is the only module that, at run time, can provide the com.example.api API.
- B. The placement of the com.example.api API in a separate module, ServiceAPI, makes it easy to install multiple provider modules.
- C. The Consumer module should require the ServiceProvider module.
- D. The ServiceProvider module should export the com.myimpl package.
- E. The ServiceProvider module does not know the identity of a module (such as Consumer) that uses the com.example.api API.

**Answer:** AC

#### NEW QUESTION 27

Given:

```
public class Main {  
    public static void main(String[] args) {  
        int i = 1;  
        for(String s : args) {  
            System.out.println((i++) + " " + s);  
        }  
    }  
}
```

executed with this command: java Main one two three  
What is the output of this class?

- A. The compilation fails.
- B. 1) one2) two3) three
- C. A java.lang.ArrayIndexOutOfBoundsException is thrown.
- D. 1) one
- E. nothing

**Answer: B**

#### NEW QUESTION 29

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;

**Answer: AB**

#### NEW QUESTION 31

Which describes an aspect of Java that contributes to high performance?

- A. Java prioritizes garbage collection.
- B. Java has a library of built-in functions that can be used to enable pipeline burst execution.
- C. Java monitors and optimizes code that is frequently executed.
- D. Java automatically parallelizes code execution.

**Answer: C**

#### NEW QUESTION 35

Given:

```
public class DNASynth {  
    int aCount;  
    int tCount;  
    int cCount;  
    int gCount;  
  
    int getACount(int aCount) {  
        return aCount;  
    }  
    int getTCount(int tCount) {  
        return this.tCount;  
    }  
    int getCCount() {  
        return getTotalCount() - this.aCount - getTCount(0) - gCount;  
    }  
    int getGCount() {  
        return getGCount();  
    }  
    int getTotalCount() {  
        return aCount + getTCount(0) + this.cCount + this.gCount;  
    }  
}
```

Which two methods facilitate valid ways to read instance fields? (Choose two.)

- A. getTCount
- B. getACount
- C. getTotalCount
- D. getCCount
- E. getGCount

Answer: CD

#### NEW QUESTION 38

Given:

```
import java.util.function.BiFunction;
public class Pair<T> {
    final BiFunction<T, T, Boolean> validator;
    T left = null;
    T right = null;
    private Pair() {
        validator=null;
    }
    Pair(BiFunction<T, T, Boolean> v, T x, T y) {
        validator = v;
        set(x, y);
    }
    void set(T x, T y) {
        if (!validator.apply(x, y)) throw new IllegalArgumentException();
        setLeft(x);
        setRight(y);
    }
    void setLeft(T x) {
        left = x;
    }
    void setRight(T y) {
        right = y;
    }
    final boolean isValid() {
        return validator.apply(left, right);
    }
}
```

It is required that if p instanceof Pair then p.isValid() returns true.

Which is the smallest set of visibility changes to insure this requirement is met?

- A. setLeft and setRight must be protected.
- B. left and right must be private.
- C. isValid must be public.
- D. left, right, setLeft, and setRight must be private.

Answer: B

#### NEW QUESTION 41

Which two are functional interfaces? (Choose two.)

- A. `@FunctionalInterface`  
`interface MyRunnable {`  
    `public void run();`  
`}`
- B. `@FunctionalInterface`  
`interface MyRunnable {`  
    `public void run();`  
    `public void call();`  
`}`
- C. `interface MyRunnable {`  
    `public default void run() {}`  
    `public void run(String s);`  
`}`
- D. `@FunctionalInterface`  
`interface MyRunnable {`  
`}`
- E. `interface MyRunnable {`  
    `@FunctionalInterface`  
    `public void run();`  
`}`

- A. Option A
- B. Option B

- C. Option C
- D. Option D
- E. Option E

**Answer:** CE

#### NEW QUESTION 44

Given:

```
public class Test {
    public static void main(String[] args) {
        AnotherClass ac = new AnotherClass();
        SomeClass sc = new AnotherClass();
        ac = sc;
        sc.methodA();
        ac.methodA();
    }
}

class SomeClass {
    public void methodA() {
        System.out.println("SomeClass#methodA()");
    }
}

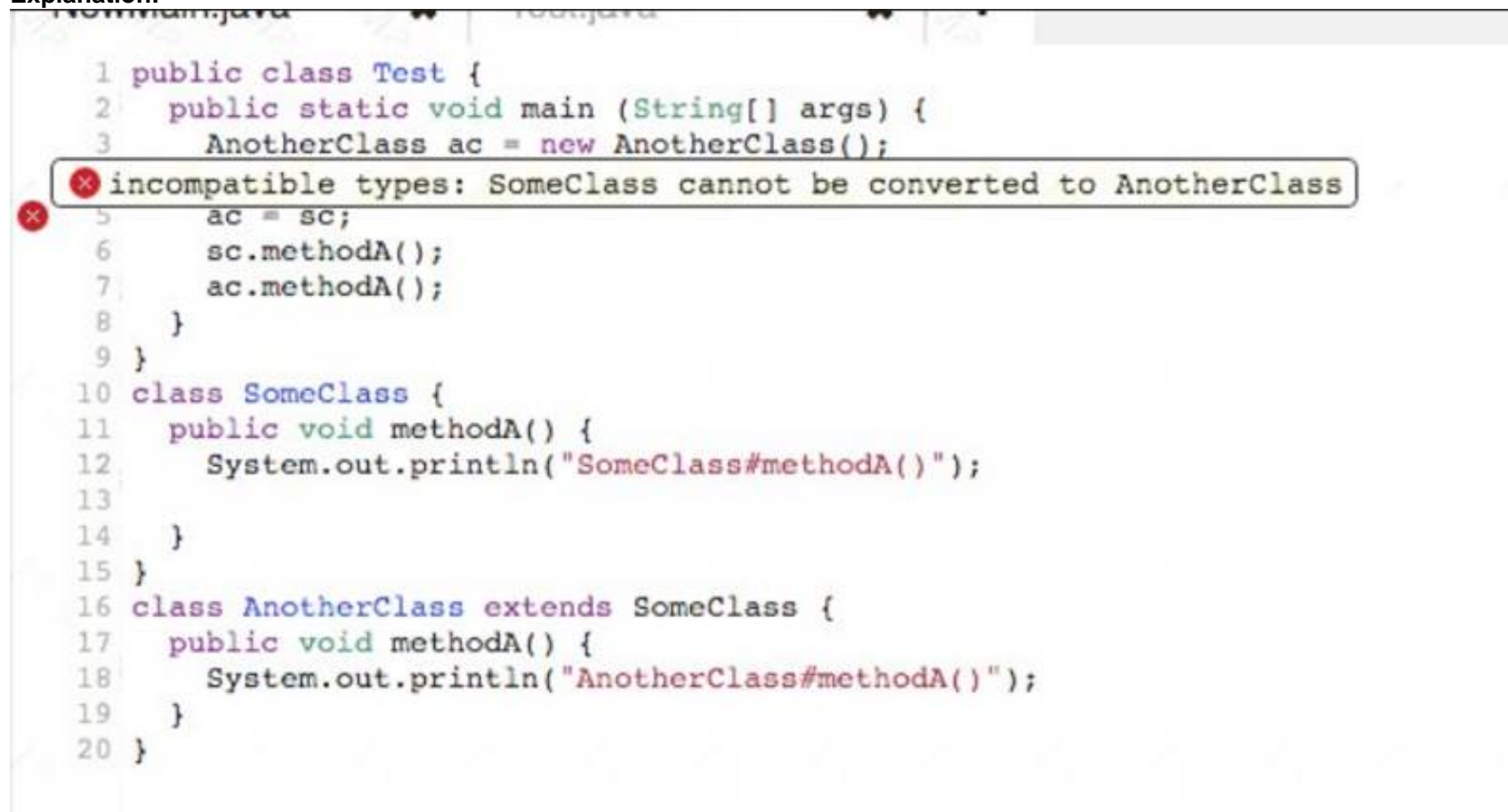
class AnotherClass extends SomeClass {
    public void methodA() {
        System.out.println("AnotherClass#methodA()");
    }
}
```

What is the result?

- A. A ClassCastException is thrown at runtime.
- B. AnotherClass#methodA()AnotherClass#methodA()
- C. The compilation fails.
- D. SomeClass#methodA()AnotherClass#methodA()
- E. AnotherClass#methodA()SomeClass#methodA()
- F. SomeClass#methodA()SomeClass#methodA()

**Answer:** C

**Explanation:**



```
1 public class Test {
2     public static void main (String[] args) {
3         AnotherClass ac = new AnotherClass();
4
5         ac = sc;
6         sc.methodA();
7         ac.methodA();
8     }
9 }
10 class SomeClass {
11     public void methodA() {
12         System.out.println("SomeClass#methodA()");
13     }
14 }
15 }
16 class AnotherClass extends SomeClass {
17     public void methodA() {
18         System.out.println("AnotherClass#methodA()");
19     }
20 }
```

#### NEW QUESTION 46

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

**Answer:** D

#### NEW QUESTION 47

Given the code fragment:

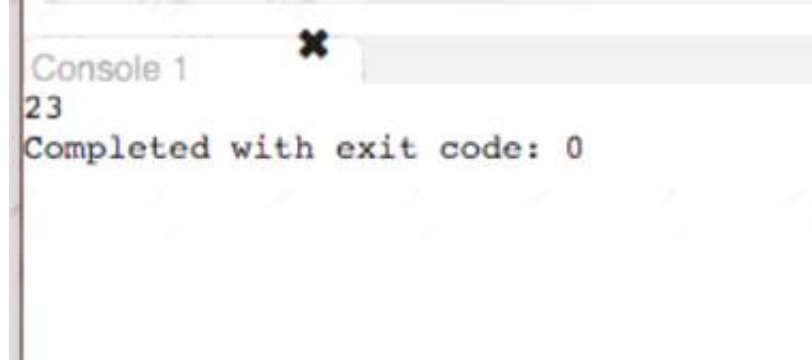
```
String s = "";  
if (Double.parseDouble("11.00f") > 11) {  
    s += 1;  
}  
if (1_7 == Integer.valueOf("17")) {  
    s += 2;  
}  
if (1024 > 1023L) {  
    s += 3;  
}  
System.out.print(s);
```

What is the result?

- A. 23
- B. 12
- C. 123
- D. 13

**Answer:** A

**Explanation:**



#### NEW QUESTION 49

Given:



```
interface MyInterface1 {
    public int method() throws Exception;
    private void pMethod() { /* an implementation of pMethod */ }
}
interface MyInterface2 {
    public static void sMethod() { /* an implementation of sMethod */ }
    public boolean equals();
}
interface MyInterface3 {
    public void method();
    public void method(String str);
}
interface MyInterface4 {
    public void dMethod() { /* an implementation of dMethod */ }
    public void method();
}
interface MyInterface5 {
    public static void sMethod();
    public void method(String str);
}
```

Which two interfaces can be used in lambda expressions? (Choose two.)

- A. MyInterface1
- B. MyInterface3
- C. MyInterface5
- D. MyInterface2
- E. MyInterface4

**Answer:** CD

#### NEW QUESTION 50

Which statement about access modifiers is correct?

- A. An instance variable can be declared with the static modifier.
- B. A local variable can be declared with the final modifier.
- C. An abstract method can be declared with the private modifier.
- D. An inner class cannot be declared with the public modifier.
- E. An interface can be declared with the protected modifier.

**Answer:** B

#### NEW QUESTION 52

Analyze the code:

```
public class Test {
    static String prefix = "Global:";
    private String name = "namespace";
    public static String getName() {
        return new Test().name;
    }
    public static void main(String[] args) {
        Test t = new Test();
        System.out.println(/* Insert code here */);
    }
}
```

Which two options can you insert inside println method to produce Global:namespace? (Choose two.)

- A. Test.prefix+Test.name
- B. new Test().prefix+new Test().name
- C. Test.prefix+Test.getName()
- D. Test.getName+prefix
- E. prefix+Test.name
- F. prefix+name

**Answer:** BC

#### NEW QUESTION 55

Given:

/code/a/Test.java containing:

```
package a;
import b.Best;
public class Test {
    public static void main(String[] args) {
        Best b = new Best();
    }
}
```

and

/code/b/Best.java containing: package b;

```
public class Best { }
```

Which is the valid way to generate bytecode for all classes?

- A. java /code/a/Test.java
- B. javac -d /code /code/a/Test
- C. java /code/a/Test.java /code/b/Best.java
- D. java -cp /code a.Test
- E. javac -d /code /code/a/Test.java /code/b/Best.java
- F. javac -d /code /code/a/Test.java

**Answer: E**

#### NEW QUESTION 56

Given:

```
public interface ExampleInterface{ }
```

Which two statements are valid to be written in this interface? (Choose two.)

- A. public abstract void methodB();
- B. final void methodG(){System.out.println("G");}
- C. private abstract void methodC();
- D. public String methodD();
- E. public int x;
- F. final void methodE();
- G. public void methodF(){System.out.println("F");}

**Answer: AD**

#### NEW QUESTION 58

Given:

```
public class Employee {
    private String name;
    private LocalDate birthday;
    // the constructors, getters, and setters methods go here
}
```

and

```
List<Employee> roster = new ArrayList<>();
// ...
Predicate<Employee> y = (Employee e) -> e.getBirthday()
    .isBefore(IsoChronology.INSTANCE.date(1989, 1, 1));
Set<String> s1 = roster.stream()
// Line 1
```

Which code fragment on line 1 makes the s1 set contain the names of all employees born before January 1, 1989?

- A. 

```
.collect(Collectors.partitioningBy(y))
    .get(true)
    .stream()
    .map(Employee::getName)
    .collect(Collectors.toCollection(TreeSet::new));
```
- B. 

```
.collect(Collectors.partitioningBy(y))
    .get(true)
    .map(Employee::getName)
    .collect(Collectors.toSet());
```
- C. 

```
.collect(Collectors.partitioningBy(y, Collectors.mapping(
    Employee::getName, Collectors.toSet())));
```
- D. 

```
.collect(Collectors.partitioningBy(y, Collectors.groupingBy(
    Employee::getName, Collectors.toCollection(TreeSet::new))));
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: B

### NEW QUESTION 63

Given:

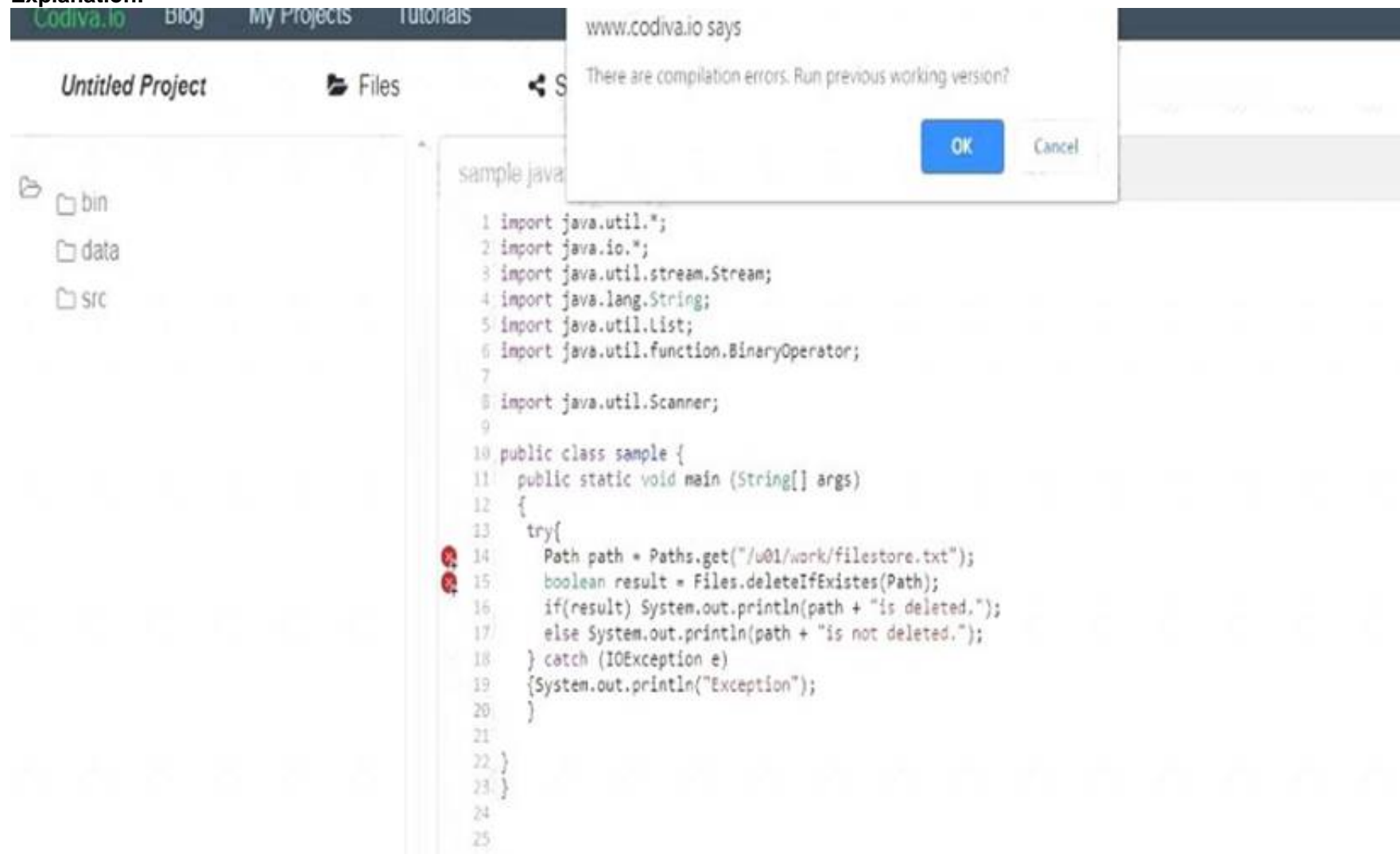
```
public class Main {
    public static void main(String[] args) {
        try {
            Path path = Paths.get("/u01/work/filestore.txt");
            boolean result = Files.deleteIfExists(path);
            if(result) System.out.println(path + "is deleted.");
            else System.out.println(path + "is not deleted.");
        } catch(IOException e) {
            System.out.println("Exception");
        }
    }
}
```

Assume the file on path does not exist. What is the result?

- A. The compilation fails.
- B. /u01/work/filestore.txt is not deleted.
- C. Exception
- D. /u01/work/filestore.txt is deleted.

Answer: A

Explanation:



### NEW QUESTION 65

Given:

```
enum Color implements Serializable {
    R(1), G(2), B(3);
    int c;
    public Color(int c) {
        this.c = c;
    }
}
```

What action ensures successful compilation?

- A. Replace public Color(int c) with private Color(int c).
- B. Replace int c; with private int c;.
- C. Replace int c; with private final int c;.
- D. Replace enum Color implements Serializable with public enum Color.
- E. Replace enum Color with public enum Color.

Answer: A



**Explanation:**

```
1
2 import java.io.*;
3 import java.util.*;
4 class Hello {
5
6
7     enum Color implements Serializable {
8         R(1), G(2), B(3);
9         int c;
10        private Color (int c) {
11            this.c = c;
12        }
13    }
14 }
```

**NEW QUESTION 69**

Given:

```
public class Over {
    public void analyze(Object[] o){
        System.out.println("I am an object array");
    }
    public void analyze(long[] l){
        System.out.println("I am an array");
    }
    public void analyze(Object o){
        System.out.println("I am an object");
    }
    public static void main(String[] args) {
        int[] nums = new int[10];
        new Over().analyze(nums); // line 1
    }
}
```

What is the output?

- A. I am an object array
- B. The compilation fails due to an error in line 1.
- C. I am an array
- D. I am an object

**Answer: D****NEW QUESTION 74**

Given:

```
List<String> list = ... ;
list.forEach( x -> { System.out.println(x); } );
```

What is the type of x?

- A. char
- B. List<Character>
- C. String
- D. List<String>

**Answer: C****NEW QUESTION 77**

Given:

```
public class Tester {
    public static void main(String[] args) {
        byte x = 7, y = 6;
        // line 1
        System.out.println(z);
    }
}
```

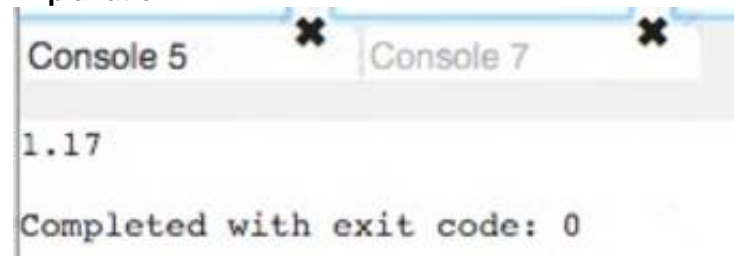
Which expression when added at line 1 will produce the output of 1.17?

- A. float z = (float)(Math.round((float)x/y\*100)/100);
- B. float z = Math.round((int)(x/y),2);
- C. float z = Math.round((float)x/y,2);

D. float z = Math.round((float)x/y\*100)/(float)100;

**Answer:** D

**Explanation:**



#### NEW QUESTION 81

Which three annotation uses are valid? (Choose three.)

- A. Function<String, String> func = (@NonNull x) > x.toUpperCase();
- B. var v = "Hello" + (@Interned) "World"
- C. Function<String, String> func = (var @NonNull x) > x.toUpperCase();
- D. Function<String, String> func = (@NonNull var x) > x.toUpperCase();
- E. var myString = (@NonNull String) str;
- F. var obj = new @Interned MyObject();

**Answer:** ACF

#### NEW QUESTION 85

Given:

```
public class Main {
    public static void main(String[] args) {
        for(int i = 0; i < args.length; i++) {
            System.out.println(i + "). " + args[i]);
            switch(args[i]) {
                case "one":
                    continue;
                case "two":
                    i--;
                    continue;
                default:
                    break;
            }
        }
    }
}
```

executed with this command: java Main one two three What is the result?

- A. 0). one
- B. 0). one1). two2). three
- C. The compilation fails.
- D. It creates an infinite loop printing:0). one1). two1). two...
- E. A java.lang.NullPointerException is thrown.

**Answer:** D

#### NEW QUESTION 86

Which code fragment compiles?



```
A. Comparator comparator = new Comparator<?>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};  
B. var comparator = new Comparator<>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};  
C. Comparator<> comparator = new Comparator<Integer>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};  
D. Comparator<Integer> comparator = new Comparator<>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer: D**

**Explanation:**

```
1 import java.io.*;  
2 import java.util.*;  
3 class abc {  
4     public static void main(String[] args) {  
5  
6         Comparator<Integer> comparator = new Comparator<>() {  
7             public int compare(Integer i, Integer j) {  
8                 return i.compareTo(j);  
9             }  
10        };  
11    }  
12 }  
13 }|  
14
```

#### NEW QUESTION 87

Given:

```
StringBuilder s = new StringBuilder("ABCD");
```

Which would cause s to be AQCD?

- A. s.replace(s.indexOf("A"), s.indexOf("C"), "Q");
- B. s.replace(s.indexOf("B"), s.indexOf("C"), "Q");
- C. s.replace(s.indexOf("B"), s.indexOf("B"), "Q");
- D. s.replace(s.indexOf("A"), s.indexOf("B"), "Q");

**Answer: B**

#### NEW QUESTION 92

.....

## Thank You for Trying Our Product

### We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### 1Z0-819 Practice Exam Features:

- \* 1Z0-819 Questions and Answers Updated Frequently
- \* 1Z0-819 Practice Questions Verified by Expert Senior Certified Staff
- \* 1Z0-819 Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* 1Z0-819 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

**100% Actual & Verified — Instant Download, Please Click**  
**[Order The 1Z0-819 Practice Test Here](#)**