



**Oracle**

**Exam Questions 1Z0-819**

Java SE 11 Developer

#### NEW 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() {

**Answer:** BC

#### NEW QUESTION 2

Which two commands are used to identify class and module dependencies? (Choose two.)

- A. jmod describe
- B. java Hello.java
- C. jdeps --list-deps
- D. jar --show-module-resolution
- E. java --show-module-resolution

**Answer:** CE

#### NEW QUESTION 3

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 4

Given:

```
public static void main(String[] args) {
    final List<String> fruits =
        List.of("Orange", "Apple", "Lemmon", "Raspberry");
    final List<String> types =
        List.of("Juice", "Pie", "Ice", "Tart");
    final var stream =
        IntStream.range(0, Math.min(fruits.size(), types.size()))
            .mapToObj((i) -> fruits.get(i) + " " + types.get(i) );
    stream. forEach(System.out::println);
}
```

What is the result?

- A. Orange Juice
- B. The compilation fails.
- C. Orange Juice Apple Pie Lemmon Ice Raspberry Tart
- D. The program prints nothing.

**Answer:** C

**Explanation:**

```
12 ~ public class Person {
13 ~     public static void main (String[] args) {
14         final List<String> fruits =
15         List.of("Orange", "Apple", "Lemmon", "raspberry");
16         final List<String> types =
17         List.of("Juice", "Pie", "Ice", "Tart");
18         final var stream =
19         IntStream.range(0, Math.min(fruits.size(), types.size()))
20         .mapToObj ((i) -> fruits.get(i) + " " + types.get(i) );
21         stream. forEach(System.out::println);
22     }
23
24 }
```

**Result**

compiled and executed in 1.227 sec(s)

```
Orange Juice
Apple Pie
Lemmon Ice
raspberry Tart
```

**NEW QUESTION 5**

Given:

```
1. {
2.     Iterator iter = List.of(1,2,3).iterator();
3.     while (iter.hasNext()) {
4.         foo(iter.next());
5.     }
6.     Iterator iter2 = List.of(1,2,3).iterator();
7.     while (iter.hasNext()) {
8.         bar(iter2.next());
9.     }
10. }
11. for (Iterator iter = List.of(1,2,3).iterator(); iter.hasNext(); ) {
12.     foo(iter.next());
13. }
14. for (Iterator iter2 = List.of(1,2,3).iterator(); iter.hasNext(); ) {
15.     bar(iter2.next());
16. }
```

Which loop incurs a compile time error?

- A. the loop starting line 11
- B. the loop starting line 7
- C. the loop starting line 14
- D. the loop starting line 3

**Answer: C**

**NEW QUESTION 6**

Given the code fragment:

```
Path currentFile = Paths.get("/scratch/exam/temp.txt"); Path outputFile = Paths.get("/scratch/exam/new.txt"); Path directory = Paths.get("/scratch/");
Files.copy(currentFile, outputFile); Files.copy(outputFile, directory);
Files.delete (outputFile);
```

The /scratch/exam/temp.txt file exists. The /scratch/exam/new.txt and /scratch/new.txt files do not exist. What is the result?

- A. /scratch/exam/new.txt and /scratch/new.txt are deleted.
- B. The program throws a FileAlreadyExistsException.
- C. The program throws a NoSuchFileException.
- D. A copy of /scratch/exam/new.txt exists in the /scratch directory and /scratch/exam/new.txt is deleted.

**Answer: C**

**Explanation:**

```
27 public class Main {
28     public static void main(String[] args) {
29         Path currentFile = Paths.get("/scratch/exam/temp.txt");
30         Path outputFile = Paths.get("/scratch/exam/new.txt");
31         Path directory = Paths.get("/scratch/");
32
33         Files.copy(currentFile, outputFile);
34         Files.copy(outputFile, directory);
35         Files.delete(outputFile);
36     }
37 }
38
```

#### NEW QUESTION 7

Given:

```
1. interface Pastry {
2.     void getIngredients();
3. }
4. abstract class Cookie implements Pastry {}
5.
6. class ChocolateCookie implements Cookie {
7.     public void getIngredients() {}
8. }
9. class CoconutChocolateCookie extends ChocolateCookie {
10.     void getIngredients(int x) {}
11. }
```

Which is true?

- A. The compilation fails due to an error in line 6.
- B. The compilation succeeds.
- C. The compilation fails due to an error in line 4.
- D. The compilation fails due to an error in line 10.
- E. The compilation fails due to an error in line 7.
- F. The compilation fails due to an error in line 9.
- G. The compilation fails due to an error in line 2.

**Answer: A**

#### NEW QUESTION 8

Given:

```
package A;
class Test {
    String name;
    public Test(String name) {
        this.name = name;
    }
    public String toString() {
        return name;
    }
}
```

and

```
package B;
import A.Test;
public class Main {
    public static void main(String[] args) {
        Test test = new Test("Student");
        System.out.println(test);
    }
}
```

What is the result?

- A. null
- B. nothing
- C. It fails to compile.
- D. java.lang.IllegalAccessException is thrown.
- E. Student

**Answer: C**

### 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

Which command line runs the main class `com.acme.Main` from the module `com.example`?

- A. `java --module-path mods com.example/com.acme.Main`
- B. `java -classpath com.example.jar com.acme.Main`
- C. `java --module-path mods -m com.example/com.acme.Main`
- D. `java -classpath com.example.jar -m com.example/com.acme.Main`

**Answer: D**

#### NEW QUESTION 11

Given:

```
class Mycar {
}
```

and

```
javac C:\workspace4\Mycar.java
```

What is the expected result of `javac`?

- A. `javac` fails to compile the class and prints the error message, `C:\workspace4\Mycar.java:1:error: package java does not exist`
- B. `javac` compiles `Mycar.java` without errors or warnings.
- C. `javac` fails to compile the class and prints the error message, `C:\workspace4\Mycar.java:1:error: expected import java.lang`
- D. `javac` fails to compile the class and prints the error message, `Error: Could not find or load main class Mycar.class`

**Answer: B**

#### NEW QUESTION 12

Given the formula to calculate a monthly mortgage payment:

$$M = P \frac{r(1+r)^n}{(1+r)^n - 1}$$

and these declarations:

```
double m;           //monthly payment
double r = 0.05/12; //monthly interest rate
int p = 100_000;     //principal
int n = 180;         //number of payments
```

How can you code the formula?

- A. `m = p * (r * Math.pow(1 + r, n) / (Math.pow(1 + r, n) - 1));`
- B. `m = p * ((r * Math.pow(1 + r, n) / (Math.pow(1 + r, n)) - 1));`
- C. `m = p * r * Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1;`
- D. `m = p * (r * Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1);`

Answer: A

#### NEW QUESTION 17

Given the code fragment:

```
var pool = Executors.newFixedThreadPool(5);
```

```
Future outcome = pool.submit(() > 1);
```

Which type of lambda expression is passed into submit()?

- A. java.lang Runnable
- B. java.util.function.Predicate
- C. java.util.function.Function
- D. java.util.concurrent.Callable

Answer: D

#### NEW QUESTION 20

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

Answer: B

#### NEW QUESTION 21

Given:

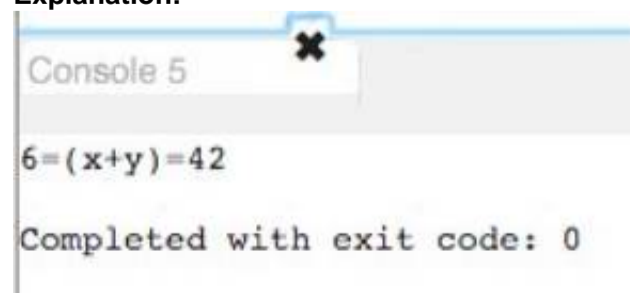
```
public class Tester {  
    public static void main(String[] args) {  
        int x = 4;  
        int y = 2;  
        System.out.println(x+y+"=(x+y)="+x+y);  
    }  
}
```

What is the result?

- A. An exception is thrown at runtime
- B. 42=(x+y)=42
- C. 42=(x+y)=6
- D. 6=(x+y)=42
- E. 6=(x+y)=6

Answer: D

Explanation:



### NEW QUESTION 23

Given:

```
class Myclass {  
    public static void main(String [] args) {  
        System.out.println(arg[1] + "--" + arg[3] + "--" + arg[0]);  
    }  
}
```

executed using this command: java Myclass My Car is red What is the output of this class?

- A. Car--red--My
- B. My--Car--is
- C. My--is--java
- D. java--Myclass--My
- E. Myclass--Car--red

**Answer:** A

### NEW QUESTION 28

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

☐ In

CommandLine Arguments

Result

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

300

NEW QUESTION 30

Given:

```
class ConSuper {
    protected ConSuper() {
        this(2);
        System.out.print("1");
    }
    protected ConSuper(int a) {
        System.out.print(a);
    }
}
```

and

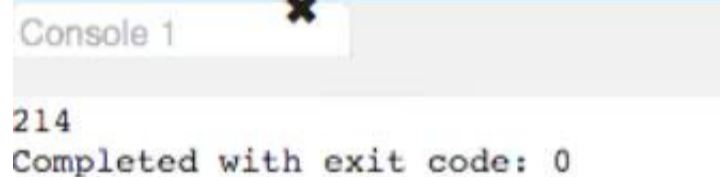
```
public class ConSub extends ConSuper{
    ConSub() {
        this(4);
        System.out.print("3");
    }
    ConSub(int a) {
        System.out.print(a);
    }
    public static void main (String[] args) {
        new ConSub(4);
    }
}
```

What is the result?

- A. 2134
- B. 2143
- C. 214
- D. 234

**Answer:** C

**Explanation:**



Console 1

214

Completed with exit code: 0

#### NEW QUESTION 34

Given:

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

and

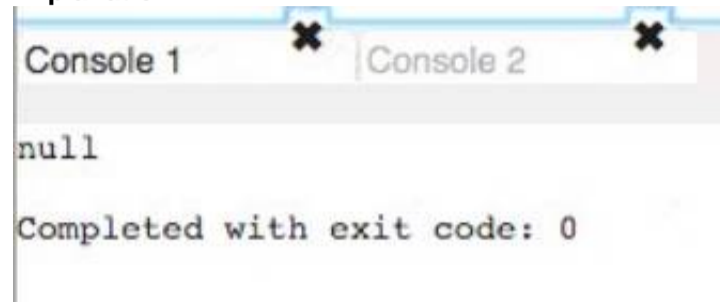
```
public class Test {
    public static void main(String args[]) {
        Person p = new Person();
        p.setName("Who");
        System.out.println(p);
    }
}
```

What is the result?

- A. D
- B. Who
- C. D
- 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 35

Given:

```
public class X {
    private Collection collection;
    public void set(Collection collection) {
        this.collection = collection;
    }
}
```

and

```
public class Y extends X {
    public void set(Map<String,String> map) {
        super.set(map); // line 1
    }
}
```

Which two lines can replace line 1 so that the Y class compiles? (Choose two.)

- A. map.forEach((k, v) -> set(v));
- B. set(map.values());
- C. super.set(List<String> map)
- D. super.set(map.values());
- E. set(map)

**Answer:** BD

### NEW QUESTION 38

Given the code fragment:

```
String s1 = new String("ORACLE");
String s2 = "ORACLE";
String s3 = s1.intern();

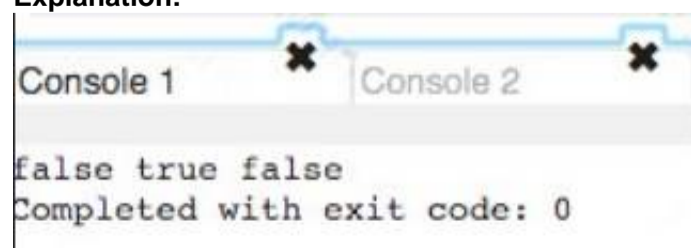
System.out.print((s1==s2) + " ");
System.out.print((s2==s3) + " ");
System.out.println(s1==s3);
```

What is the result?

- A. false true true
- B. true false false
- C. false false true
- D. false true false

**Answer:** D

**Explanation:**



```
Console 1 [X] Console 2 [X]
false true false
Completed with exit code: 0
```

#### NEW QUESTION 41

Given the code fragment:

```
char[][] arrays = {{'a', 'd'}, {'b', 'e'}, {'c', 'f'}};
for (char[] xx : arrays) {
    for (char yy : xx) {
        System.out.print(yy);
    }
    System.out.print(" ");
}
```

What is the result?

- A. ab cd ef
- B. An `ArrayIndexOutOfBoundsException` is thrown at runtime.
- C. The compilation fails.
- D. abc def
- E. ad be cf

**Answer: E**

#### NEW QUESTION 46

Given:

```
class Employee {
    String office;
}
```

and the code fragment:

```
5. public class HRApp {
6.     var employee = new ArrayList<Employee>();
7.     public var display() {
8.         var employee = new Employee();
9.         var offices = new ArrayList<>();
10.        offices.add("Chicago");
11.        offices.add("Bangalore");
12.        for (var office : offices) {
13.            System.out.print("Employee Location"+ office);
14.        }
15.    }
16. }
```

Which two lines cause compilation errors? (Choose two.)

- A. line 12
- B. line 6
- C. line 9
- D. line 8
- E. line 7

**Answer: BE**

#### NEW QUESTION 47

Given:

```
Integer[] intArray = {2, 1, 3, 4, 5};
List<Integer> list =
new ArrayList<>(Arrays.asList (intArray));
list.parallelStream()
    .forEach(e -> System.out.print(e + " "));
```

Which two are correct? (Choose two.)

- A. The output will be exactly 2 1 3 4 5.
- B. The program prints 1 4 2 3, but the order is unpredictable.
- C. Replacing `forEach()` with `forEachOrdered()`, the program prints 2 1 3 4 5, but the order is unpredictable.
- D. Replacing `forEach()` with `forEachOrdered()`, the program prints 1 2 3 4 5.
- E. Replacing `forEach()` with `forEachOrdered()`, the program prints 2 1 3 4 5.

**Answer: BD**

**Explanation:**

```

8 public class Secret {
9     public static void main(String[] args) {
10         Integer[] intArray = {1, 2, 3, 4, 5};
11         List<Integer> list =
12             new ArrayList<> (Arrays.asList (intArray));
13         list.parallelStream()
14             .forEachOrdered(e -> System.out.print(e + " "));
15     }
16 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.32 sec(s), Memory: 37040 kilobyte(s)

1 2 3 4 5

#### NEW QUESTION 50

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 53

Given the code fragment:

```

int x = 0;
do {
    x++;
    if (x == 1) {
        continue;
    }
    System.out.println(x);
} while(x < 1);

```

What is the result?

- A. 01
- B. 1
- C. The program prints nothing.
- D. It prints 1 in the infinite loop.

**Answer:** D

#### NEW QUESTION 57

Which statement about a functional interface is true?

- A. It must be defined with the public access modifier.
- B. It must be annotated with `@FunctionalInterface`.
- C. It is declared with a single abstract method.
- D. It is declared with a single default method.
- E. It cannot have any private methods and static methods.

**Answer:** C

#### NEW QUESTION 61

Given:

```
String originalPath = "data\\projects\\a-project\\..\\..\\another-project"; Path path = Paths.get(originalPath); System.out.print(path.normalize());
```

What is the result?

- A. data\\another-project
- B. data\\projects\\a-project\\another-project
- C. data\\projects\\a-project\\..\\..\\another-project
- D. data\\projects\\a-project\\..\\..\\another-project

**Answer:** D

**Explanation:**

```
1 import java.util.*;
2 import java.io.*;
3 import java.nio.file.*;
4
5 public class Test {
6
7     public static void main(String[] args) {
8         String originalPath = "data\\projects\\a-project\\..\\..\\another-project";
9         Path path = Paths.get(originalPath);
10        System.out.print(path.normalize());
11    }
12 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4 ☐ Interactive Stdin Input

CommandLine Arguments

**Result**  
CPU Time: 0.19 sec(s), Memory: 31984 kilobyte(s)

```
data\\projects\\a-project\\..\\..\\another-project
```

#### NEW QUESTION 64

Given:

```
public class Person {
    private String name = "Joe Bloggs";
    public Person(String name) {
        this.name = name;
    }
    public String toString() {
        return name;
    }
}
```

and

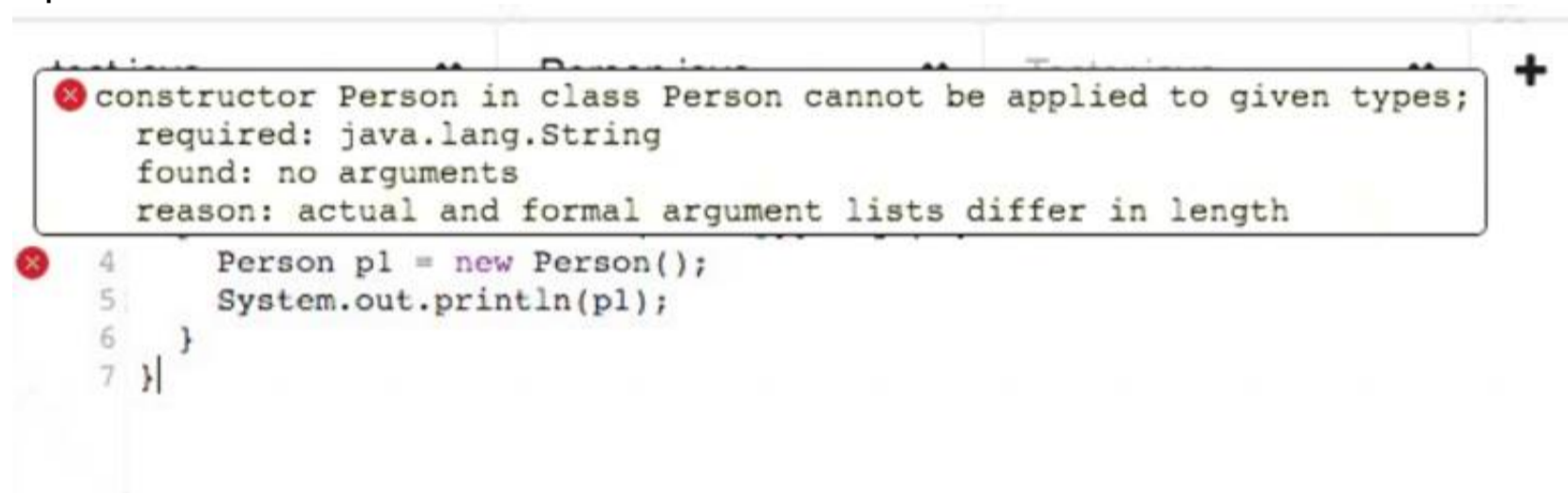
```
public class Tester {
    public static void main(String[] args) {
        Person p1 = new Person(); // line 1
        System.out.println(p1);
    }
}
```

What is the result?

- A. null
- B. Joe Bloggs
- C. The compilation fails due to an error in line 1.
- D. p1

**Answer: C**

**Explanation:**



#### NEW QUESTION 66

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 71

Given:

```
public class Main {
    public static void main(String[] args) {
        Consumer consumer = msg -> System.out::print; // line 1
        consumer.accept("Hello Lambda !");
    }
}
```

This code results in a compilation error.

Which code should be inserted on line 1 for a successful compilation?

- A. Consumer consumer = msg -> { return System.out.print(msg); };
- B. Consumer consumer = var arg -> {System.out.print(arg);};
- C. Consumer consumer = (String args) -> System.out.print(args);
- D. Consumer consumer = System.out::print;

**Answer: D**

**Explanation:**

```

1  import java.util.*;
2  import java.io.*;
3  import java.nio.file.*;
4  import java.util.List;
5  import java.util.function.Consumer;
6
7  public class Main {
8
9      public static void main(String[] args) {
10         Consumer consumer = System.out::print;
11         consumer.accept("Hello Lambda !");
12     }
13 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32896 kilobyte(s)

Hello Lambda !

#### NEW QUESTION 75

Given:

```

1. public class Main {
2.     public static void greet(String... args) {
3.         System.out.print("Hello ");
4.         for (String arg : args) {
5.             System.out.println(arg);
6.         }
7.     }
8.     public static void main(String[] args) {
9.         Main c = null;
10.        c.greet();
11.    }
12. }

```

What is the result?

- A. NullPointerException is thrown at line 4.
- B. NullPointerException is thrown at line 10.
- C. A compilation error occurs.
- D. Hello

Answer: D

Explanation:



Console 4 Console 5

hello

Completed with exit code: 0

#### NEW QUESTION 76

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

- A. Modular jars loaded from --module-path are automatic modules.
- B. Any named module can directly access all classes in an automatic module.
- C. Classes found in -classpath are part of an unnamed module.
- D. Modular jars loaded from -classpath are automatic modules.
- E. If a package is defined in both the named module and the unnamed module, then the package in the unnamed module is ignored.

**Answer:** AC

#### NEW QUESTION 78

Given this requirement:

Module vehicle depends on module part and makes its com.vehicle package available for all other modules. Which module-info.java declaration meets the requirement?

**A**

```
module vehicle{  
    requires part;  
    exports com.vehicle;  
}
```

**B**

```
module vehicle {  
    requires part;  
    uses com.vehicle;  
}
```

**C**

```
module vehicle{  
    requires part;  
    exports com.vehicle to part;  
}
```

**D**

```
module vehicle {  
    requires com.vehicle;  
    exports part;  
}
```

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

**Answer:** A

#### NEW QUESTION 80

Given:

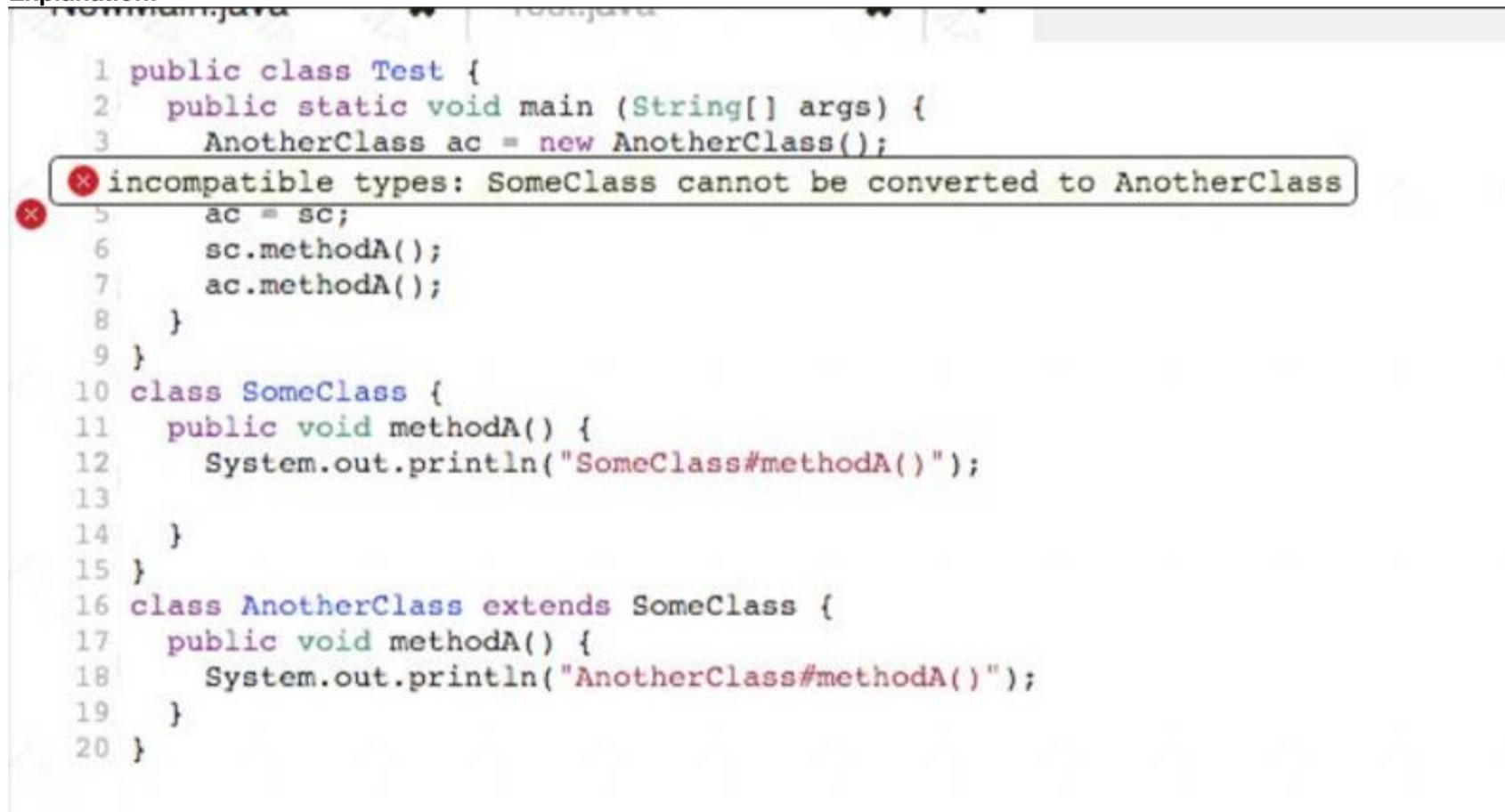
```
public class Test {
    public static void main(String[] args) {
        AnotherClass ac = new AnotherClass();
        SomeClass sc = new AnotherClass();
        ac = sc;
        sc.methodA();
        ac.methodA();
    }
}
class SomeClass {
    public void methodA() {
        System.out.println("SomeClass#methodA()");
    }
}
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:**



#### NEW QUESTION 81

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 84

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 87

Given:

```
public class Employee {
    private String name;
    private String locality;
    /* the constructor, getter and setter methods code goes here */
}
```

and:

```
8. List<Employee> roster = new ArrayList<>();
9. long empCount = roster.stream()
10. /* insert code here */
11. System.out.print(empCount);
```

Which code, when inserted on line 10, prints the number of unique localities from the roster list?

- A. .map(Employee::getLocality).distinct().count();
- B. map(e -> e.getLocality()).count();
- C. .map(e -> e.getLocality()).collect(Collectors.toSet()).count();
- D. .filter(Employee::getLocality).distinct().count();

**Answer:** D

#### NEW QUESTION 89

Given:

```
public class Main {
    public static void main(String[] args) {
        var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
        Optional<Integer> result = numbers.stream().filter(x -> x % 3 != 0).reduce((i, j)
-> i + j);
        result.ifPresent(System.out::print); // line 1
    }
}
```

Which is true about line 1?

- A. If the value is not present, a NoSuchElementException is thrown at run time.
- B. It always executes the System.out::print statement.
- C. If the value is not present, a NullPointerException is thrown at run time.
- D. If the value is not present, nothing is done.

**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 import java.util.Optional;
11
12
13 public class Main {
14     public static void main(String[] args) {
15         var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
16         Optional<Integer> result = numbers.stream().filter (x -> x % 3 != 0).reduce( (i, j) -> i + j);
17     }
18 }
19 }

```

Result

CPU Time: 0.18 sec(s), Memory: 33380 kilobyte(s)

JDoodle in Action.... Running the program...

#### NEW QUESTION 91

Given:

```

import java.io.*;
public class Tester {
    public static void main(String[] args) {
        try {
            doA();
            doB();
        } catch (IOException e) {
            System.out.print("c");
            return;
        } finally{
            System.out.print("d");
        }
        System.out.print("f");
    }
    private static void doA() {
        System.out.print("a");
        if (false) {
            throw new IndexOutOfBoundsException();
        }
    }
    private static void doB() throws FileNotFoundException {
        System.out.print("b");
        if (true) {
            throw new FileNotFoundException();
        }
    }
}

```

What is the result?

- A. The compilation fails.
- B. abdf
- C. abd
- D. adf
- E. abcd

Answer: E

#### NEW QUESTION 95

Given:

```
package test.t1;
public class A {
    public int x = 42;
    protected A() {}           // line 1
}
```

and

```
package test.t2;
import test.t1.*;
public class B extends A {
    int x = 17;                 // line 2
    public B() { super(); }     // line 3
}
```

and

```
package test;
import test.t1.*;
import test.t2.*;
public class Tester {
    public static void main(String[] args) {
        A obj = new B();        // line 4
        System.out.println(obj.x); // line 5
    }
}
```

What is the result?

- A. 42
- B. The compilation fails due to an error in line 4.
- C. 17
- D. The compilation fails due to an error in line 3.
- E. The compilation fails due to an error in line 2.
- F. The compilation fails due to an error in line 1.
- G. The compilation fails due to an error in line 5.

**Answer:** A

#### NEW QUESTION 100

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:** AD

#### NEW QUESTION 104

Given the contents:

MessageBundle.properties file: message=Hello MessageBundle\_en.properties file: message=Hello (en) MessageBundle\_US.properties file: message=Hello (US)

MessageBundle\_en\_US.properties file: message=Hello (en\_US) MessageBundle\_fr\_FR.properties file: message=Bonjour

and the code fragment: Locale.setDefault(Locale.FRANCE);

Locale currentLocale = new Locale.Builder().setLanguage("en").build();

ResourceBundle messages = ResourceBundle.getBundle("MessageBundle", currentLocale); System.out. println(messages.getString("message"));

Which file will display the content on executing the code fragment?

- A. MessageBundle\_en\_US.properties
- B. MessageBundle\_en.properties
- C. MessageBundle\_fr\_FR.properties
- D. MessageBundle\_US.properties
- E. MessageBundle.properties

**Answer:** C

#### NEW QUESTION 107

Given:

```
for(var i = 0; i < 10; i++) {  
    switch(i%5) {  
        case 2:  
            i *= i;  
            break;  
        case 3:  
            i++;  
            break;  
        case 1:  
        case 4:  
            i++;  
            continue;  
        default:  
            break;  
    }  
    System.out.print(i + " ");  
    i++;  
}
```

What is the result?

- A. nothing
- B. 10
- C. 0 4 9

**Answer:** A

#### NEW QUESTION 112

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. 1Null null
- D. 111
- E. A ClassCastException is thrown at run time.

**Answer:** B

**Explanation:**

```
Console 4  
Error: Could not find or load main class CustomType  
Caused by: java.lang.ClassNotFoundException: CustomType
```

#### NEW QUESTION 115

Given this enum declaration:

```
1.enum Alphabet {  
2.    A, B, C  
3.  
4. }
```

Examine this code: `System.out.println(Alphabet.getFirstLetter());`  
What code should be written at line 3 to make this code print A?

- A. `final String getFirstLetter() { return A.toString(); }`
- B. `static String getFirstLetter() { return Alphabet.values()[1].toString(); }`
- C. `static String getFirstLetter() { return A.toString(); }`
- D. `String getFirstLetter() { return A.toString(); }`

**Answer:** C

#### NEW QUESTION 120

.....

## About ExamBible

*[Your Partner of IT Exam](#)*

## Found in 1998

ExamBible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, ExamBible has its unique advantages that other companies could not achieve.

## Our Advances

### \* 99.9% Uptime

All examinations will be up to date.

### \* 24/7 Quality Support

We will provide service round the clock.

### \* 100% Pass Rate

Our guarantee that you will pass the exam.

### \* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

#### NEW 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() {

**Answer:** BC

#### NEW QUESTION 2

Which two commands are used to identify class and module dependencies? (Choose two.)

- A. jmod describe
- B. java Hello.java
- C. jdeps --list-deps
- D. jar --show-module-resolution
- E. java --show-module-resolution

**Answer:** CE

#### NEW QUESTION 3

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 4

Given:

```
public static void main(String[] args) {
    final List<String> fruits =
        List.of("Orange", "Apple", "Lemmon", "Raspberry");
    final List<String> types =
        List.of("Juice", "Pie", "Ice", "Tart");
    final var stream =
        IntStream.range(0, Math.min(fruits.size(), types.size()))
            .mapToObj((i) -> fruits.get(i) + " " + types.get(i) );
    stream. forEach(System.out::println);
}
```

What is the result?

- A. Orange Juice
- B. The compilation fails.
- C. Orange Juice Apple Pie Lemmon Ice Raspberry Tart
- D. The program prints nothing.

**Answer:** C

**Explanation:**

```
12 ~ public class Person {
13 ~     public static void main (String[] args) {
14         final List<String> fruits =
15         List.of("Orange", "Apple", "Lemmon", "raspberry");
16         final List<String> types =
17         List.of("Juice", "Pie", "Ice", "Tart");
18         final var stream =
19         IntStream.range(0, Math.min(fruits.size(), types.size()))
20         .mapToObj ((i) -> fruits.get(i) + " " + types.get(i) );
21         stream. forEach(System.out::println);
22     }
23
24 }
```

**Result**

compiled and executed in 1.227 sec(s)

```
Orange Juice
Apple Pie
Lemmon Ice
raspberry Tart
```

**NEW QUESTION 5**

Given:

```
1. {
2.     Iterator iter = List.of(1,2,3).iterator();
3.     while (iter.hasNext()) {
4.         foo(iter.next());
5.     }
6.     Iterator iter2 = List.of(1,2,3).iterator();
7.     while (iter.hasNext()) {
8.         bar(iter2.next());
9.     }
10. }
11. for (Iterator iter = List.of(1,2,3).iterator(); iter.hasNext(); ) {
12.     foo(iter.next());
13. }
14. for (Iterator iter2 = List.of(1,2,3).iterator(); iter.hasNext(); ) {
15.     bar(iter2.next());
16. }
```

Which loop incurs a compile time error?

- A. the loop starting line 11
- B. the loop starting line 7
- C. the loop starting line 14
- D. the loop starting line 3

**Answer: C**

**NEW QUESTION 6**

Given the code fragment:

```
Path currentFile = Paths.get("/scratch/exam/temp.txt"); Path outputFile = Paths.get("/scratch/exam/new.txt"); Path directory = Paths.get("/scratch/");
Files.copy(currentFile, outputFile); Files.copy(outputFile, directory);
Files.delete (outputFile);
```

The /scratch/exam/temp.txt file exists. The /scratch/exam/new.txt and /scratch/new.txt files do not exist. What is the result?

- A. /scratch/exam/new.txt and /scratch/new.txt are deleted.
- B. The program throws a FileAlreadyExistsException.
- C. The program throws a NoSuchFileException.
- D. A copy of /scratch/exam/new.txt exists in the /scratch directory and /scratch/exam/new.txt is deleted.

**Answer: C**

**Explanation:**

```
27 public class Main {
28     public static void main(String[] args) {
29         Path currentFile = Paths.get("/scratch/exam/temp.txt");
30         Path outputFile = Paths.get("/scratch/exam/new.txt");
31         Path directory = Paths.get("/scratch/");
32
33         Files.copy(currentFile, outputFile);
34         Files.copy(outputFile, directory);
35         Files.delete(outputFile);
36     }
37 }
38
```

#### NEW QUESTION 7

Given:

```
1. interface Pastry {
2.     void getIngredients();
3. }
4. abstract class Cookie implements Pastry {}
5.
6. class ChocolateCookie implements Cookie {
7.     public void getIngredients() {}
8. }
9. class CoconutChocolateCookie extends ChocolateCookie {
10.     void getIngredients(int x) {}
11. }
```

Which is true?

- A. The compilation fails due to an error in line 6.
- B. The compilation succeeds.
- C. The compilation fails due to an error in line 4.
- D. The compilation fails due to an error in line 10.
- E. The compilation fails due to an error in line 7.
- F. The compilation fails due to an error in line 9.
- G. The compilation fails due to an error in line 2.

**Answer: A**

#### NEW QUESTION 8

Given:

```
package A;
class Test {
    String name;
    public Test(String name) {
        this.name = name;
    }
    public String toString() {
        return name;
    }
}
```

and

```
package B;
import A.Test;
public class Main {
    public static void main(String[] args) {
        Test test = new Test("Student");
        System.out.println(test);
    }
}
```

What is the result?

- A. null
- B. nothing
- C. It fails to compile.
- D. java.lang.IllegalAccessException is thrown.
- E. Student

**Answer: C**

### 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.compareTo(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.compareTo(p2.name);}` on line 2.
- D. Insert `Comparator<Person>` on line 1. Insert `public int compare(Person person) { return person.name.compareTo(this.name);}` on line 2.

**Answer: B**

#### NEW QUESTION 10

Which command line runs the main class `com.acme.Main` from the module `com.example`?

- A. `java --module-path mods com.example/com.acme.Main`
- B. `java -classpath com.example.jar com.acme.Main`
- C. `java --module-path mods -m com.example/com.acme.Main`
- D. `java -classpath com.example.jar -m com.example/com.acme.Main`

**Answer: D**

#### NEW QUESTION 11

Given:

```
class Mycar {
}
```

and

```
javac C:\workspace4\Mycar.java
```

What is the expected result of `javac`?

- A. `javac` fails to compile the class and prints the error message, `C:\workspace4\Mycar.java:1:error: package java does not exist`
- B. `javac` compiles `Mycar.java` without errors or warnings.
- C. `javac` fails to compile the class and prints the error message, `C:\workspace4\Mycar.java:1:error: expected import java.lang`
- D. `javac` fails to compile the class and prints the error message, `Error: Could not find or load main class Mycar.class`

**Answer: B**

#### NEW QUESTION 12

Given the formula to calculate a monthly mortgage payment:

$$M = P \frac{r(1+r)^n}{(1+r)^n - 1}$$

and these declarations:

```
double m;           //monthly payment
double r = 0.05/12; //monthly interest rate
int p = 100_000;     //principal
int n = 180;         //number of payments
```

How can you code the formula?

- A. `m = p * (r * Math.pow(1 + r, n) / (Math.pow(1 + r, n) - 1));`
- B. `m = p * ((r * Math.pow(1 + r, n) / (Math.pow(1 + r, n)) - 1));`
- C. `m = p * r * Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1;`
- D. `m = p * (r * Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1);`

Answer: A

#### NEW QUESTION 17

Given the code fragment:

```
var pool = Executors.newFixedThreadPool(5);
```

```
Future outcome = pool.submit(() > 1);
```

Which type of lambda expression is passed into submit()?

- A. java.lang.Runnable
- B. java.util.function.Predicate
- C. java.util.function.Function
- D. java.util.concurrent.Callable

Answer: D

#### NEW QUESTION 20

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 21

Given:

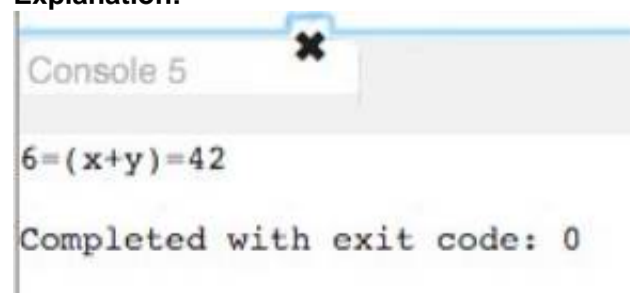
```
public class Tester {  
    public static void main(String[] args) {  
        int x = 4;  
        int y = 2;  
        System.out.println(x+y+"=(x+y)="+x+y);  
    }  
}
```

What is the result?

- A. An exception is thrown at runtime
- B. 42=(x+y)=42
- C. 42=(x+y)=6
- D. 6=(x+y)=42
- E. 6=(x+y)=6

Answer: D

Explanation:



### NEW QUESTION 23

Given:

```
class Myclass {  
    public static void main(String [] args) {  
        System.out.println(arg[1] + "--" + arg[3] + "--" + arg[0]);  
    }  
}
```

executed using this command: java Myclass My Car is red What is the output of this class?

- A. Car--red--My
- B. My--Car--is
- C. My--is--java
- D. java--Myclass--My
- E. Myclass--Car--red

**Answer:** A

### NEW QUESTION 28

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



In

CommandLine Arguments

Result

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

300

#### NEW QUESTION 30

Given:

```
class ConSuper {
    protected ConSuper() {
        this(2);
        System.out.print("1");
    }
    protected ConSuper(int a) {
        System.out.print(a);
    }
}
```

and

```
public class ConSub extends ConSuper{
    ConSub() {
        this(4);
        System.out.print("3");
    }
    ConSub(int a) {
        System.out.print(a);
    }
    public static void main (String[] args) {
        new ConSub(4);
    }
}
```

What is the result?

- A. 2134
- B. 2143
- C. 214
- D. 234

**Answer:** C

**Explanation:**

Console 1

214  
Completed with exit code: 0

#### NEW QUESTION 34

Given:

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

and

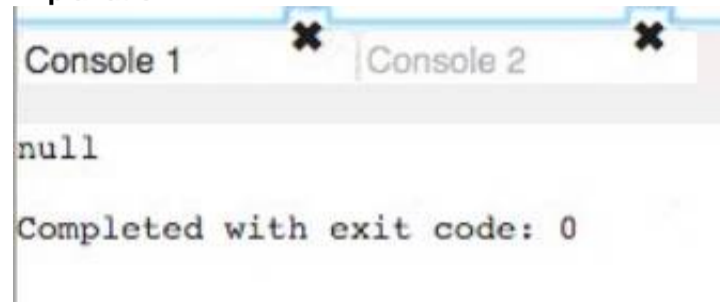
```
public class Test {
    public static void main(String args[]) {
        Person p = new Person();
        p.setName("Who");
        System.out.println(p);
    }
}
```

What is the result?

- A. D
- B. Who
- C. D
- 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 35

Given:

```
public class X {
    private Collection collection;
    public void set(Collection collection) {
        this.collection = collection;
    }
}
```

and

```
public class Y extends X {
    public void set(Map<String,String> map) {
        super.set(map); // line 1
    }
}
```

Which two lines can replace line 1 so that the Y class compiles? (Choose two.)

- A. map.forEach((k, v) -> set(v));
- B. set(map.values());
- C. super.set(List<String> map)
- D. super.set(map.values());
- E. set(map)

**Answer:** BD

### NEW QUESTION 38

Given the code fragment:

```
String s1 = new String("ORACLE");
String s2 = "ORACLE";
String s3 = s1.intern();

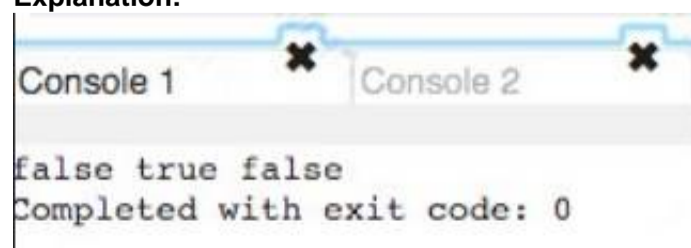
System.out.print((s1==s2) + " ");
System.out.print((s2==s3) + " ");
System.out.println(s1==s3);
```

What is the result?

- A. false true true
- B. true false false
- C. false false true
- D. false true false

**Answer:** D

**Explanation:**



```
Console 1 [X] Console 2 [X]
false true false
Completed with exit code: 0
```

#### NEW QUESTION 41

Given the code fragment:

```
char[][] arrays = {{'a', 'd'}, {'b', 'e'}, {'c', 'f'}};
for (char[] xx : arrays) {
    for (char yy : xx) {
        System.out.print(yy);
    }
    System.out.print(" ");
}
```

What is the result?

- A. ab cd ef
- B. An `ArrayIndexOutOfBoundsException` is thrown at runtime.
- C. The compilation fails.
- D. abc def
- E. ad be cf

**Answer: E**

#### NEW QUESTION 46

Given:

```
class Employee {
    String office;
}
```

and the code fragment:

```
5. public class HRApp {
6.     var employee = new ArrayList<Employee>();
7.     public var display() {
8.         var employee = new Employee();
9.         var offices = new ArrayList<>();
10.        offices.add("Chicago");
11.        offices.add("Bangalore");
12.        for (var office : offices) {
13.            System.out.print("Employee Location"+ office);
14.        }
15.    }
16. }
```

Which two lines cause compilation errors? (Choose two.)

- A. line 12
- B. line 6
- C. line 9
- D. line 8
- E. line 7

**Answer: BE**

#### NEW QUESTION 47

Given:

```
Integer[] intArray = {2, 1, 3, 4, 5};
List<Integer> list =
new ArrayList<>(Arrays.asList (intArray));
list.parallelStream()
    .forEach(e -> System.out.print(e + " "));
```

Which two are correct? (Choose two.)

- A. The output will be exactly 2 1 3 4 5.
- B. The program prints 1 4 2 3, but the order is unpredictable.
- C. Replacing `forEach()` with `forEachOrdered()`, the program prints 2 1 3 4 5, but the order is unpredictable.
- D. Replacing `forEach()` with `forEachOrdered()`, the program prints 1 2 3 4 5.
- E. Replacing `forEach()` with `forEachOrdered()`, the program prints 2 1 3 4 5.

**Answer: BD**

**Explanation:**

```

8 public class Secret {
9     public static void main(String[] args) {
10         Integer[] intArray = {1, 2, 3, 4, 5};
11         List<Integer> list =
12             new ArrayList<> (Arrays.asList (intArray));
13         list.parallelStream()
14             .forEachOrdered(e -> System.out.print(e + " "));
15     }
16 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.32 sec(s), Memory: 37040 kilobyte(s)

1 2 3 4 5

#### NEW QUESTION 50

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 53

Given the code fragment:

```

int x = 0;
do {
    x++;
    if (x == 1) {
        continue;
    }
    System.out.println(x);
} while(x < 1);

```

What is the result?

- A. 01
- B. 1
- C. The program prints nothing.
- D. It prints 1 in the infinite loop.

**Answer:** D

#### NEW QUESTION 57

Which statement about a functional interface is true?

- A. It must be defined with the public access modifier.
- B. It must be annotated with `@FunctionalInterface`.
- C. It is declared with a single abstract method.
- D. It is declared with a single default method.
- E. It cannot have any private methods and static methods.

**Answer:** C

#### NEW QUESTION 61

Given:

```
String originalPath = "data\\projects\\a-project\\..\\..\\another-project"; Path path = Paths.get(originalPath); System.out.print(path.normalize());
```

What is the result?

- A. data\\another-project
- B. data\\projects\\a-project\\another-project
- C. data\\projects\\a-project\\..\\..\\another-project
- D. data\\projects\\a-project\\..\\..\\another-project

**Answer:** D

**Explanation:**

```
1 import java.util.*;
2 import java.io.*;
3 import java.nio.file.*;
4
5 public class Test {
6
7     public static void main(String[] args) {
8         String originalPath = "data\\projects\\a-project\\..\\..\\another-project";
9         Path path = Paths.get(originalPath);
10        System.out.print(path.normalize());
11    }
12 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4 ☐ Interactive Stdin Input

CommandLine Arguments

...

**Result**  
CPU Time: 0.19 sec(s), Memory: 31984 kilobyte(s)

```
data\\projects\\a-project\\..\\..\\another-project
```

#### NEW QUESTION 64

Given:

```
public class Person {
    private String name = "Joe Bloggs";
    public Person(String name) {
        this.name = name;
    }
    public String toString() {
        return name;
    }
}
```

and

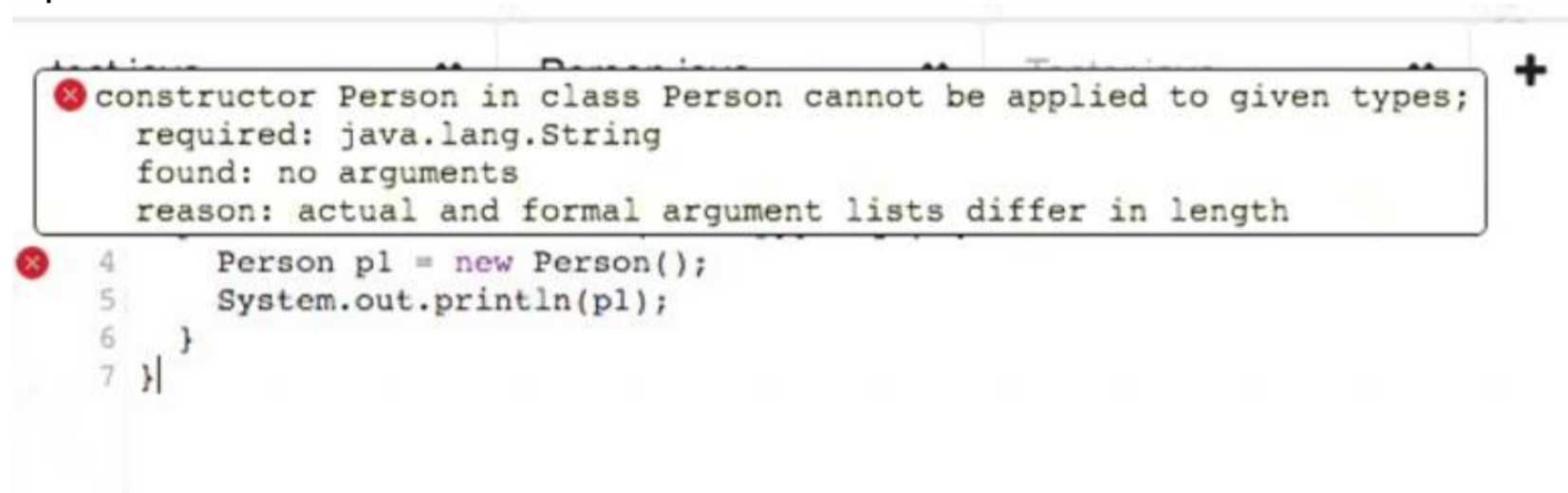
```
public class Tester {
    public static void main(String[] args) {
        Person p1 = new Person(); // line 1
        System.out.println(p1);
    }
}
```

What is the result?

- A. null
- B. Joe Bloggs
- C. The compilation fails due to an error in line 1.
- D. p1

**Answer: C**

**Explanation:**



#### NEW QUESTION 66

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 71

Given:

```
public class Main {
    public static void main(String[] args) {
        Consumer consumer = msg -> System.out::print; // line 1
        consumer.accept("Hello Lambda !");
    }
}
```

This code results in a compilation error.

Which code should be inserted on line 1 for a successful compilation?

- A. Consumer consumer = msg -> { return System.out.print(msg); };
- B. Consumer consumer = var arg -> {System.out.print(arg);};
- C. Consumer consumer = (String args) -> System.out.print(args);
- D. Consumer consumer = System.out::print;

**Answer: D**

**Explanation:**

```

1  import java.util.*;
2  import java.io.*;
3  import java.nio.file.*;
4  import java.util.List;
5  import java.util.function.Consumer;
6
7  public class Main {
8
9      public static void main(String[] args) {
10         Consumer consumer = System.out::print;
11         consumer.accept("Hello Lambda !");
12     }
13 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32896 kilobyte(s)

Hello Lambda !

#### NEW QUESTION 75

Given:

```

1. public class Main {
2.     public static void greet(String... args) {
3.         System.out.print("Hello ");
4.         for (String arg : args) {
5.             System.out.println(arg);
6.         }
7.     }
8.     public static void main(String[] args) {
9.         Main c = null;
10.        c.greet();
11.    }
12. }

```

What is the result?

- A. NullPointerException is thrown at line 4.
- B. NullPointerException is thrown at line 10.
- C. A compilation error occurs.
- D. Hello

Answer: D

Explanation:



#### NEW QUESTION 76

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

- A. Modular jars loaded from --module-path are automatic modules.
- B. Any named module can directly access all classes in an automatic module.
- C. Classes found in -classpath are part of an unnamed module.
- D. Modular jars loaded from -classpath are automatic modules.
- E. If a package is defined in both the named module and the unnamed module, then the package in the unnamed module is ignored.

**Answer:** AC

#### NEW QUESTION 78

Given this requirement:

Module vehicle depends on module part and makes its com.vehicle package available for all other modules. Which module-info.java declaration meets the requirement?

**A**

```
module vehicle{  
    requires part;  
    exports com.vehicle;  
}
```

**B**

```
module vehicle {  
    requires part;  
    uses com.vehicle;  
}
```

**C**

```
module vehicle{  
    requires part;  
    exports com.vehicle to part;  
}
```

**D**

```
module vehicle {  
    requires com.vehicle;  
    exports part;  
}
```

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

**Answer:** A

#### NEW QUESTION 80

Given:

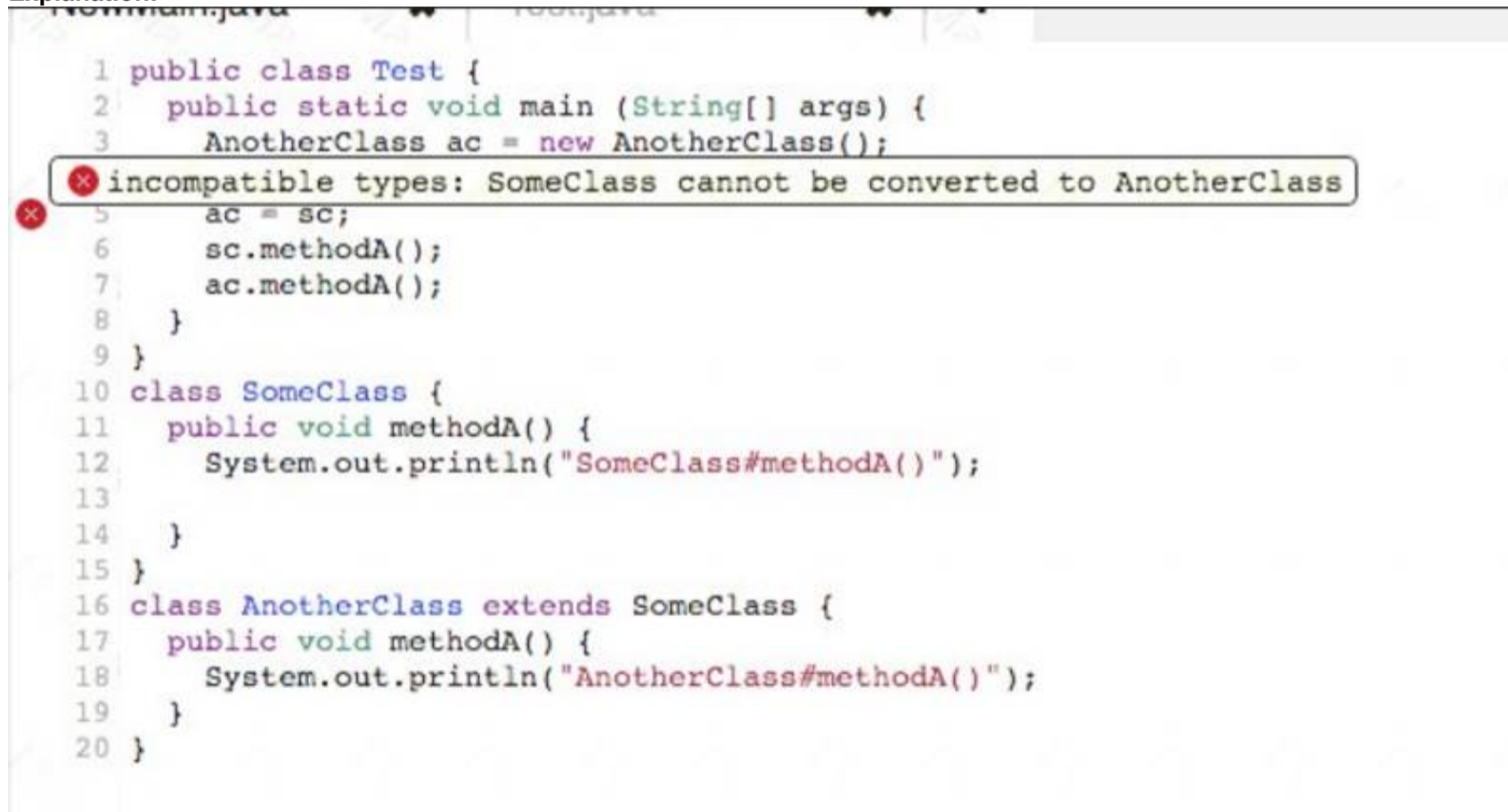
```
public class Test {
    public static void main(String[] args) {
        AnotherClass ac = new AnotherClass();
        SomeClass sc = new AnotherClass();
        ac = sc;
        sc.methodA();
        ac.methodA();
    }
}
class SomeClass {
    public void methodA() {
        System.out.println("SomeClass#methodA()");
    }
}
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 81

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 84

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 87

Given:

```
public class Employee {
    private String name;
    private String locality;
    /* the constructor, getter and setter methods code goes here */
}
```

and:

```
8. List<Employee> roster = new ArrayList<>();
9. long empCount = roster.stream()
10. /* insert code here */
11. System.out.print(empCount);
```

Which code, when inserted on line 10, prints the number of unique localities from the roster list?

- A. .map(Employee::getLocality).distinct().count();
- B. map(e -> e.getLocality()).count();
- C. .map(e -> e.getLocality()).collect(Collectors.toSet()).count();
- D. .filter(Employee::getLocality).distinct().count();

**Answer:** D

#### NEW QUESTION 89

Given:

```
public class Main {
    public static void main(String[] args) {
        var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
        Optional<Integer> result = numbers.stream().filter(x -> x % 3 != 0).reduce((i, j)
-> i + j);
        result.ifPresent(System.out::print); // line 1
    }
}
```

Which is true about line 1?

- A. If the value is not present, a NoSuchElementException is thrown at run time.
- B. It always executes the System.out::print statement.
- C. If the value is not present, a NullPointerException is thrown at run time.
- D. If the value is not present, nothing is done.

**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 import java.util.Optional;
11
12
13 public class Main {
14     public static void main(String[] args) {
15         var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
16         Optional<Integer> result = numbers.stream().filter (x -> x % 3 != 0).reduce( (i, j) -> i + j);
17     }
18 }
19 }

```

Result

CPU Time: 0.18 sec(s), Memory: 33380 kilobyte(s)

JDoodle in Action.... Running the program...

#### NEW QUESTION 91

Given:

```

import java.io.*;
public class Tester {
    public static void main(String[] args) {
        try {
            doA();
            doB();
        } catch (IOException e) {
            System.out.print("c");
            return;
        } finally{
            System.out.print("d");
        }
        System.out.print("f");
    }
    private static void doA() {
        System.out.print("a");
        if (false) {
            throw new IndexOutOfBoundsException();
        }
    }
    private static void doB() throws FileNotFoundException {
        System.out.print("b");
        if (true) {
            throw new FileNotFoundException();
        }
    }
}

```

What is the result?

- A. The compilation fails.
- B. abdf
- C. abd
- D. adf
- E. abcd

Answer: E

#### NEW QUESTION 95

Given:

```
package test.t1;
public class A {
    public int x = 42;
    protected A() {}           // line 1
}
```

and

```
package test.t2;
import test.t1.*;
public class B extends A {
    int x = 17;                 // line 2
    public B() { super(); }     // line 3
}
```

and

```
package test;
import test.t1.*;
import test.t2.*;
public class Tester {
    public static void main(String[] args) {
        A obj = new B();        // line 4
        System.out.println(obj.x); // line 5
    }
}
```

What is the result?

- A. 42
- B. The compilation fails due to an error in line 4.
- C. 17
- D. The compilation fails due to an error in line 3.
- E. The compilation fails due to an error in line 2.
- F. The compilation fails due to an error in line 1.
- G. The compilation fails due to an error in line 5.

**Answer:** A

#### NEW QUESTION 100

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:** AD

#### NEW QUESTION 104

Given the contents:

MessageBundle.properties file: message=Hello MessageBundle\_en.properties file: message=Hello (en) MessageBundle\_US.properties file: message=Hello (US)

MessageBundle\_en\_US.properties file: message=Hello (en\_US) MessageBundle\_fr\_FR.properties file: message=Bonjour

and the code fragment: Locale.setDefault(Locale.FRANCE);

Locale currentLocale = new Locale.Builder().setLanguage("en").build();

ResourceBundle messages = ResourceBundle.getBundle("MessageBundle", currentLocale); System.out. println(messages.getString("message"));

Which file will display the content on executing the code fragment?

- A. MessageBundle\_en\_US.properties
- B. MessageBundle\_en.properties
- C. MessageBundle\_fr\_FR.properties
- D. MessageBundle\_US.properties
- E. MessageBundle.properties

**Answer:** C

#### NEW QUESTION 107

Given:

```
for(var i = 0; i < 10; i++) {  
    switch(i%5) {  
        case 2:  
            i *= i;  
            break;  
        case 3:  
            i++;  
            break;  
        case 1:  
        case 4:  
            i++;  
            continue;  
        default:  
            break;  
    }  
    System.out.print(i + " ");  
    i++;  
}
```

What is the result?

- A. nothing
- B. 10
- C. 0 4 9

**Answer:** A

#### NEW QUESTION 112

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

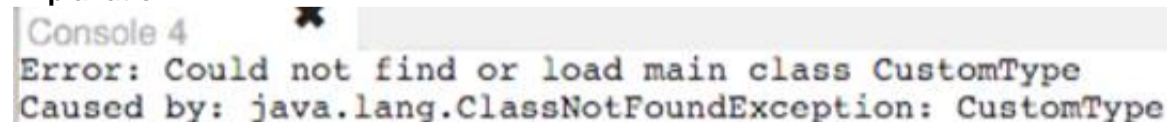
```
public class Test extends CustomType {  
    public static void main(String[] args) {  
        String[] words = {"banana","orange","apple","lemon"};  
        Integer[] numbers = {1,2,3,4,5};  
        CustomType type = new CustomType();  
        CustomType<String> stringType = new CustomType<>();  
        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. 1Null null
- D. 111
- E. A ClassCastException is thrown at run time.

**Answer:** B

**Explanation:**



Console 4  
Error: Could not find or load main class CustomType  
Caused by: java.lang.ClassNotFoundException: CustomType

#### NEW QUESTION 115

Given this enum declaration:

```
1.enum Alphabet {  
2.    A, B, C  
3.  
4. }
```

Examine this code: `System.out.println(Alphabet.getFirstLetter());`  
What code should be written at line 3 to make this code print A?

- A. `final String getFirstLetter() { return A.toString(); }`
- B. `static String getFirstLetter() { return Alphabet.values()[1].toString(); }`
- C. `static String getFirstLetter() { return A.toString(); }`
- D. `String getFirstLetter() { return A.toString(); }`

**Answer:** C

#### NEW QUESTION 120

.....

## Relate Links

**100% Pass Your 1Z0-819 Exam with ExamBible Prep Materials**

<https://www.exambible.com/1Z0-819-exam/>

## Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - <https://www.exambible.com/>